# EVALUATION OF THE ANTHROPOMETRY SYSTEM

Michael E. Pittman, Ph.D.

June 14, 1990

Naval Biodynamics Laboratory
P.O. Box 29407
New Orleans, LA 70189-0407



**Approved for public release; distribution is unlimited.**

Prepared for

**Naval Medical Research and Development Command
Bethesda, MD 20889-5044**

# EVALUATION OF THE ANTHROPOMETRY SYSTEM

**Michael E. Pittman, Ph.D.**
*Mathematical Physics*
*Scientific & Engineering Programming*
57 Emile Avenue
Kenner, Louisiana 70065
(504) 443-2355

June 14, 1990

Final Report on Contract: P. O. Number: N00205-90-M-D005

## Contract Statement

**Evaluation of the Present NavBioDynLab X-ray Anthropometry System. Analysis, exposition, enhancement and documentation of the x-ray digitization and 3-D reconstruction algorithms, development of error analysis code for incorporation into digitization and reconstruction algorithms.**

## Method

After examining the current x-ray digitization process and estimating the errors involved with these methods, with the collaboration of C. J. Mugnier, it was decided to bring in one of the top photogrammetric computer programs, *GIANT*. GIANT has a built-in error-propagation capability but it needed to be converted for use at NBDL for anthropometry and possibly for future use with the high-speed photo system. GIANT was developed on main-frame computers and is in current use in many areas of the world on VAX systems and would have to be converted for use on the HP/UNIX system. As an aid in the conversion, compilations and test runs were made on a PC version in addition. Other pre- and post-processing routines were brought in as needed (GHOSH, PREP) and modified for NBDL's needs or written in-house (TPLATE, ANTHRO). Major modifications were needed for GIANT itself to function in the new environment (HP & PC) and to suit the needs of the application.

## Results

Minimal control exists in object space with too few object points being digitized. This causes larger errors than in most photogrammetric systems where camera stations and object points are determined by a least-square adjustment of very highly over-determined systems. To reduce errors in object space, such as the coordinates of the t-plate, below about 5mm required writing an ancillary program to constrain the three coordinates on the t-plate externally after the first pass with GIANT and to use those constrained values in a second pass. Resulting errors were shrunk to about 1mm typically.

A series of acceleration runs were made with primates where the x-rays were taken in a different corner (mirror image through the sagittal plane). Problems with the existing anthropometry program caused an error in the location of the t-plate which defied efforts to correct. Much concern over the loss of use of the entire series of primate runs prompted the correction of these as the first priority. A sample run using the 'bad corner' is enclosed as Appendix 1 (the 80-column format for output was also developed here). The t-plate is in its correct location (positive z). Errors in this one-pass sample are rather high and can be reduced to about 1mm using the distance constraints on the t-plate.

Appendix 2 contains a sample run of PREP, the pre-processor program for GIANT. The measured plate coordinates must be converted into a plate-centered system and corrections made for radial lens distortion (if enough fiducial coordinates are measured).

Appendix 3 contains the full source code listings for all the software used and some of the developmental tools such as subroutine flow diagrams for GIANT and PREP.

## Future Efforts

**1: X-Ray Anthropometry**

Using these programs, the next phase should be to continue the successful analysis and rescue of faulty data on primate x-rays, to automate the use of these rigorous photogrammetric tools and to train the technical staff in their use.

**2: Assessment and Evaluation of the High-Speed Photogrammetric System**

These programs could be used to perform a system analysis on the existing high-speed PDS equipment and mensuration techniques. An error budget should be developed based on photogrammetric error propagation in order to assess the order of precision of the current system and to establish technical specifications for any contemplated system upgrade.

**3: Develop a Graphical Interface for PREP and GIANT**

A major portion of photogrammetric analysis is editing for data quality. Since all measurements are related to position and attitude, visualization of data and data errors is critical to efficient and effective analysis.

**4: Dynamic Camera Calibration**

Current high-speed camera calibration is based on a simple "bench test." State-of-the-art photogrammetric analysis allows for post-block dynamic camera calibration based on using plate residuals obtained from the current data set and instrumentation (camera) under actual dynamic stress. The software can and should be enhanced with this capability.

**5: 3-D Vector Constraints in Object Space**

Control in object space is limited to constraining the positions of object points and camera stations. The program can be made much more useful to NBDL if it had the capability to constrain distances between object points (such as various targets on the t-plate) or between camera stations, and to assure coplanarity between selected object points.

# Appendix 1

# PC Giant

## Sample Run

## 14 June 1990

# File: OPT.DAT Options Data File for Giant Sample Run in 'Bad Corner'

```
Rhesus X-RAY X-corner[22.5 Deg Rotation w/o T-PLATE HELD]
01001000001119000 1 1
0.000250  0.000250  0.000250
 AP  CAM     -1820.09
 LAT CAM     -1118.14
********
  A/P      -0.368        0.242         1.875     0.25     0.25     0.25
  A/P  -23946.476 -150509.816    -5323.691  10000.   10000.   10000.
  LAT       0.905        0.141         0.453     0.25     0.25     0.25
  LAT   10905.677   723520.744  -10010.193  10000.   10000.   10000.
********
   1      0.2347      0.0508      0.0972    0.001    0.001    0.001
   2      0.0469      0.0508      0.0194    0.001    0.001    0.001
   3      0.0469      0.2540      0.0194    0.001    0.001    0.001
   4      0.2347      0.2540      0.0972    0.001    0.001    0.001
   5     -0.0972      0.2540      0.2347    0.001    0.001    0.001
   6     -0.0972      0.0508      0.2347    0.001    0.001    0.001
   7     -0.0233      0.0508      0.0563    0.001    0.001    0.001
   8     -0.0194      0.2540      0.0469    0.001    0.001    0.001
   9      0.2152      0.1524      0.1441
  10      0.0825      0.1524      0.1991
  11     -0.0503      0.1524      0.2541
  12      0.0825      0.0508      0.1991
  13      0.0825      0.2540      0.1991
********
```

# File: IMG.DAT Image Data File for Giant Sample Run in 'Bad Corner'

```
A/P                       3.00      3.00 AP   CAM

        3    -86.0552  104.4956  Photo   A/P
       13     -0.3556  114.1476  Photo   A/P
        4    124.0536  105.7910  Photo   A/P
       11   -134.0612   -0.8128  Photo   A/P
       10      0.0000    0.0000  Photo   A/P
        9    120.7262    0.3810  Photo   A/P
        2    -85.0646 -105.4862  Photo   A/P
       12      0.3556 -113.7412  Photo   A/P
        1    123.8504 -106.2228  Photo   A/P
      lam     68.3006   83.5660  Photo   A/P
      ram    -23.4188   58.9534  Photo   A/P
      lon     33.1470   85.2932  Photo   A/P
      ron     -3.2766   74.5490  Photo   A/P
      ctp     11.5570  135.7122  Photo   A/P
      ltp     84.9630  133.9088  Photo   A/P
      rtp    -52.2986  125.9840  Photo   A/P
********
LAT                       3.00      3.00 LAT  CAM

        5    -99.3648  106.7054  Photo   LAT
       13     -0.4572  120.0912  Photo   LAT
        8    114.0714  107.8738  Photo   LAT
       11   -103.6574   -0.6096  Photo   LAT
       10      0.0000    0.0000  Photo   LAT
        9    122.4026   -0.6350  Photo   LAT
        6    -97.2566 -106.8070  Photo   LAT
       12      0.7874 -121.0818  Photo   LAT
        7    105.2322 -107.4166  Photo   LAT
      lam     18.3896  152.2222  Photo   LAT
      ram     18.3896  152.2222  Photo   LAT
      lon    -59.3090  147.5994  Photo   LAT
      ron    -54.9402  135.9154  Photo   LAT
      ctp    -43.5102  195.8086  Photo   LAT
      ltp     29.3624  211.8614  Photo   LAT
      rtp     34.0360  193.8274  Photo   LAT
********
```

Object Space Reference System is Rectangular

Rotation Angles are Object-to-Photo

Complete Triangulation process is requested

Error Propagation is requested

[Eigenvector/Eigenvalue output]

Unit Variance will be based on constrained camera parameters

All Image Residuals will be listed

Triangulated Object Coordinates will be saved

Adjusted Camera Station Parameters will be saved

esus X-RAY X-corner[22.5 Deg Rotation w/o T-PLATE HELD]

FRAME A/P

PRINCIPAL DISTANCE =-1820.0900 mm
Std. Dev. of X = 3.0000 mm
Std. Dev. of Y = 3.0000 mm

CAMERA STATION PARAMETERS

| P O S I T I O N | Std. Dev. | A T T I T U D E (Object to Photo) | Std. Dev. |
|---|---|---|---|
| = -0.3680 m | 0.2500 m | OMEGA = - 02 39 46.4760 | 01 00 0.0000 |
| = 0.2420 m | 0.2500 m | PHI = - 15 05 9.8160 | 01 00 0.0000 |
| Z = 1.8750 m | 0.2500 m | KAPPA = - 00 53 23.6910 | 01 00 0.0000 |

PLATE COORDINATES in millimeters

| ID | X | Y | ID | X | Y |
|---|---|---|---|---|---|
| 3 | -86.0552 | 104.4956 | 13 | -0.3556 | 114.1476 |
| 4 | 124.0536 | 105.7910 | 11 | -134.0612 | -0.8128 |
| 10 | 0.0000 | 0.0000 | 9 | 120.7262 | 0.3810 |
| 2 | -85.0646 | -105.4862 | 12 | 0.3556 | -113.7412 |
| 1 | 123.8504 | -106.2228 | lam | 68.3006 | 83.5660 |
| ram | -23.4188 | 58.9534 | lon | 33.1470 | 85.2932 |
| ron | -3.2766 | 74.5490 | ctp | 11.5570 | 135.7122 |
| ltp | 84.9630 | 133.9088 | rtp | -52.2986 | 125.9840 |

FRAME LAT

PRINCIPAL DISTANCE =-1118.1400 mm
Std. Dev. of X = 3.0000 mm
Std. Dev. of Y = 3.0000 mm

CAMERA STATION PARAMETERS

| P O S I T I O N | | Std. Dev. | A T T I T U D E (Object to Photo) | | Std. Dev. |
|---|---|---|---|---|---|
| X = | 0.9050 m | 0.2500 m | OMEGA = | 01 09 5.6770 | 01 00 0.0000 |
| Y = | 0.1410 m | 0.2500 m | PHI = | 72 35 20.7440 | 01 00 0.0000 |
| Z = | 0.4530 m | 0.2500 m | KAPPA = | - 01 00 10.1930 | 01 00 0.0000 |

PLATE COORDINATES in millimeters

| ID | X | Y | ID | X | Y |
|---|---|---|---|---|---|
| 5 | -99.3648 | 106.7054 | 13 | -0.4572 | 120.0912 |
| 8 | 114.0714 | 107.8738 | 11 | -103.6574 | -0.6096 |
| 10 | 0.0000 | 0.0000 | 9 | 122.4026 | -0.6350 |
| 6 | -97.2566 | -106.8070 | 12 | 0.7874 | -121.0818 |
| 7 | 105.2322 | -107.4166 | lam | 18.3896 | 152.2222 |
| ram | 18.3896 | 152.2222 | lon | -59.3090 | 147.5994 |
| ron | -54.9402 | 135.9154 | ctp | -43.5102 | 195.8086 |
| ltp | 29.3624 | 211.8614 | rtp | 34.0360 | 193.8274 |

C A M E R A   S T A T I O N S   C O R R E C T I O N S

| | ------- P O S I T I O N ------- | | | --------- A T T I T U D E ------ | | |
|---|---|---|---|---|---|---|
| | X | Y | Z | Omega | Phi | Kappa |
| | | | Iteration 1 | | | |
| A/P | 0.0305 | 0.0103 | -0.0026 m. | -0.006236 | 0.016230 | -0.001716 |
| LAT | 0.0067 | -0.0021 | 0.0024 m. | 0.008362 | -0.000160 | -0.007329 |

Provisional Weighted Sum of Squares = 618.471

| | | | Iteration 2 | | | |
|---|---|---|---|---|---|---|
| A/P | -0.0001 | -0.0021 | 0.0062 m. | 0.001256 | 0.000835 | -0.000132 |
| LAT | -0.0030 | 0.0003 | -0.0018 m. | -0.000481 | 0.000813 | 0.000539 |

Provisional Weighted Sum of Squares = 516.804

| | | | Iteration 3 | | | |
|---|---|---|---|---|---|---|
| A/P | -0.0001 | 0.0004 | 0.0001 m. | -0.000193 | -0.000066 | 0.000154 |
| LAT | 0.0003 | 0.0000 | 0.0001 m. | 0.000051 | -0.000023 | -0.000081 |

Provisional Weighted Sum of Squares = 516.886

# T R I A N G U L A T E D   I M A G E   P O I N T S   R E S I D U A L S

(in micrometers)

| | | | |
|---|---|---|---|
| 3 | *0* | A/P | |
| | | 7139 | |
| | | -9379 | |
| 13 | *0* | A/P | LAT |
| | | -2735 | -3916 |
| | | -19368 | 18880 |
| 4 | *0* | A/P | |
| | | -6477 | |
| | | -9660 | |
| 11 | *0* | A/P | LAT |
| | | 3952 | -6127 |
| | | -17030 | 6796 |
| 10 | *0* | A/P | LAT |
| | | -2569 | -4675 |
| | | -11958 | 7810 |
| 9 | *0* | A/P | LAT |
| | | -7884 | 8531 |
| | | -7034 | 10516 |
| 2 | *0* | A/P | |
| | | 7440 | |
| | | 4698 | |
| 12 | *0* | A/P | LAT |
| | | -2428 | -5769 |
| | | -4338 | -2539 |
| 1 | *0* | A/P | |
| | | -6052 | |
| | | 3433 | |
| lam | | A/P | LAT |
| | | 1034 | -28 |
| | | 7672 | -5725 |

## T R I A N G U L A T E D   I M A G E   P O I N T S   R E S I D U A L S

(in micrometers)

| | A/P | LAT |
|---|---|---|
| ram | 3472 | -293 |
| | 29232 | -24064 |
| lon | 896 | -38 |
| | 6867 | -5501 |
| ron | 1194 | -108 |
| | 10175 | -8460 |
| ctp | 353 | 22 |
| | 1994 | -1631 |
| ltp | 368 | 28 |
| | 1936 | -1417 |
| rtp | 2088 | 99 |
| | 12534 | -10612 |

| | LAT |
|---|---|
| 5 *0* | 180 |
| | 8187 |
| 8 *0* | 8312 |
| | 8631 |
| 6 *0* | -2299 |
| | 1142 |
| 7 *0* | 5821 |
| | -217 |

| | | |
|---|---|---|
| Weighted Sum of Squares (Camera) | = | 1.4 |
| Weighted Sum of Squares (Object) | = | 9.0 |
| Weighted Sum of Squares (Plates) | = | 496.4 |
| Weighted Sum of Squares (Total) | = | 506.8 |
| Degrees of Freedom............. | = | 43 |

a posteriori Variance of Unit Weight = 11.786

## T R I A N G U L A T E D   C A M E R A   S T A T I O N S
(Object to Photo)

| Ident | | Position | Error Ellipsoid | | | ---> | Length |
|---|---|---|---|---|---|---|---|
| | X = | -0.3377 m. | -0.7394 | +0.2585 | -0.6216 | ---> | 0.0634 m. |
| A/P | Y = | 0.2506 m. | -0.3623 | -0.9310 | +0.0438 | ---> | 0.0562 m. |
| | Z = | 1.8786 m. | -0.5674 | +0.2577 | +0.7821 | ---> | 0.0458 m. |

| | | | Std Dev |
|---|---|---|---|
| Attitude: | Omega =- | 02 57 33.4999 | 01 52 6.2746 |
| | Phi =- | 14 06 43.5429 | 01 56 2.0447 |
| | Kappa =- | 00 59 13.2256 | 01 28 33.8278 |

| Ident | | Position | Error Ellipsoid | | | ---> | Length |
|---|---|---|---|---|---|---|---|
| | X = | 0.9089 m. | +0.3387 | -0.0549 | +0.9393 | ---> | 0.0275 m. |
| LAT | Y = | 0.1391 m. | +0.9392 | -0.0394 | -0.3410 | ---> | 0.0209 m. |
| | Z = | 0.4537 m. | +0.0557 | +0.9977 | +0.0383 | ---> | 0.0117 m. |

| | | | Std Dev |
|---|---|---|---|
| Attitude: | Omega = | 01 36 21.6690 | 02 24 17.2658 |
| | Phi = | 72 37 30.6338 | 01 34 11.5321 |
| | Kappa =- | 01 23 47.5096 | 02 15 17.2301 |

## U M M A R Y   S T A T I S T I C S   F O R   C A M E R A   S T A T I O N S

RMS For Standard Deviations

| | | | | |
|---|---|---|---|---|
| Count = 2 | X = | 0.0434 m. | Omega = | 02 09 12.1294 |
| | Y = | 0.0405 m. | Phi = | 01 45 40.7366 |
| | Z = | 0.0422 m. | Kappa = | 01 54 20.2547 |

## T R I A N G U L A T E D   O B J E C T   P O I N T S

| Ident | | Position (meters) | | Error Ellipsoid ---> | | | Length (m) |
|---|---|---|---|---|---|---|---|
| | | X = | -0.0236 | +9.164E-01 | +8.687E-02 | +3.907E-01 | 0.0034 |
| 7 | *0* | Y = | 0.0508 | -3.976E-01 | +8.488E-02 | +9.136E-01 | 0.0033 |
| | | Z = | 0.0570 | +4.621E-02 | -9.926E-01 | +1.123E-01 | 0.0033 |
| | | X = | -0.0971 | +9.735E-01 | +8.551E-02 | +2.119E-01 | 0.0034 |
| 6 | *0* | Y = | 0.0507 | -2.190E-01 | +8.424E-02 | +9.721E-01 | 0.0033 |
| | | Z = | 0.2344 | +6.527E-02 | -9.928E-01 | +1.007E-01 | 0.0033 |
| | | X = | -0.0199 | +9.101E-01 | -1.126E-01 | +3.988E-01 | 0.0034 |
| 8 | *0* | Y = | 0.2529 | +3.773E-01 | -1.728E-01 | -9.098E-01 | 0.0033 |
| | | Z = | 0.0478 | +1.714E-01 | +9.785E-01 | -1.147E-01 | 0.0033 |
| | | X = | -0.0973 | +9.711E-01 | -1.109E-01 | +2.114E-01 | 0.0034 |
| 5 | *0* | Y = | 0.2530 | +2.286E-01 | +1.779E-01 | -9.571E-01 | 0.0033 |
| | | Z = | 0.2347 | +6.853E-02 | +9.778E-01 | +1.981E-01 | 0.0033 |
| | | X = | 0.0467 | +8.065E-01 | -1.724E-01 | +5.655E-01 | 0.0116 |
| rtp | | Y = | 0.2965 | +5.648E-01 | -5.829E-02 | -8.232E-01 | 0.0095 |
| | | Z = | 0.1552 | +1.749E-01 | +9.833E-01 | +5.034E-02 | 0.0074 |
| | | X = | 0.1696 | +9.024E-01 | -1.774E-01 | +3.927E-01 | 0.0115 |
| ltp | | Y = | 0.2930 | +3.905E-01 | -4.892E-02 | -9.193E-01 | 0.0092 |
| | | Z = | 0.2013 | +1.823E-01 | +9.829E-01 | +2.513E-02 | 0.0071 |
| | | X = | 0.0888 | +9.393E-01 | -1.740E-01 | +2.957E-01 | 0.0113 |
| ctp | | Y = | 0.2938 | +2.979E-01 | -1.429E-02 | -9.545E-01 | 0.0094 |
| | | Z = | 0.2322 | +1.703E-01 | +9.847E-01 | +3.839E-02 | 0.0072 |
| | | X = | 0.0739 | +9.554E-01 | -1.176E-01 | +2.709E-01 | 0.0110 |
| ron | | Y = | 0.2446 | +2.700E-01 | -2.370E-02 | -9.626E-01 | 0.0091 |
| | | Z = | 0.2373 | +1.196E-01 | +9.928E-01 | +9.108E-03 | 0.0070 |
| | | X = | 0.1055 | +9.644E-01 | -1.272E-01 | +2.320E-01 | 0.0112 |
| lon | | Y = | 0.2513 | +2.312E-01 | -2.114E-02 | -9.727E-01 | 0.0091 |
| | | Z = | 0.2488 | +1.287E-01 | +9.916E-01 | +9.026E-03 | 0.0069 |
| | | X = | 0.0713 | +8.755E-01 | -1.183E-01 | +4.685E-01 | 0.0110 |
| ram | | Y = | 0.2478 | +4.667E-01 | -4.440E-02 | -8.833E-01 | 0.0089 |
| | | Z = | 0.1770 | +1.253E-01 | +9.920E-01 | +1.636E-02 | 0.0070 |
| | | X = | 0.1528 | +9.185E-01 | -1.259E-01 | +3.748E-01 | 0.0111 |
| lam | | Y = | 0.2503 | +3.727E-01 | -4.122E-02 | -9.271E-01 | 0.0089 |
| | | Z = | 0.2038 | +1.321E-01 | +9.912E-01 | +9.045E-03 | 0.0068 |
| | | X = | 0.2353 | -3.041E-01 | +1.060E-01 | +9.467E-01 | 0.0034 |
| 1 | *0* | Y = | 0.0504 | +5.888E-01 | -7.603E-01 | +2.743E-01 | 0.0033 |
| | | Z = | 0.0974 | +7.489E-01 | +6.408E-01 | +1.688E-01 | 0.0033 |
| | | X = | 0.0825 | +8.938E-01 | +1.075E-01 | +4.354E-01 | 0.0009 |
| 12 | *0* | Y = | 0.0509 | -4.431E-01 | +6.134E-02 | +8.944E-01 | 0.0009 |
| | | Z = | 0.1991 | +6.947E-02 | -9.923E-01 | +1.025E-01 | 0.0009 |

T R I A N G U L A T E D   O B J E C T   P O I N T S

| Ident | | | Position (meters) | Error Ellipsoid ---> | | | Length (m) |
|---|---|---|---|---|---|---|---|
| | | X = | 0.0461 | -2.014E-01 | +1.045E-01 | +9.739E-01 | 0.0034 |
| 2 | *0* | Y = | 0.0503 | -6.492E-01 | -7.588E-01 | -5.284E-02 | 0.0033 |
| | | Z = | 0.0193 | -7.335E-01 | +6.429E-01 | -2.207E-01 | 0.0033 |
| | | X = | 0.2152 | +7.725E-01 | +1.020E-03 | +6.350E-01 | 0.0009 |
| 9 | *0* | Y = | 0.1523 | -6.347E-01 | +3.093E-02 | +7.721E-01 | 0.0009 |
| | | Z = | 0.1442 | -1.885E-02 | -9.995E-01 | +2.454E-02 | 0.0009 |
| | | X = | 0.0825 | +9.193E-01 | -1.003E-02 | +3.935E-01 | 0.0009 |
| 10 | *0* | Y = | 0.1524 | -3.923E-01 | +5.607E-02 | +9.181E-01 | 0.0009 |
| | | Z = | 0.1991 | -3.127E-02 | -9.984E-01 | +4.761E-02 | 0.0009 |
| | | X = | -0.0503 | +9.762E-01 | -3.936E-04 | +2.166E-01 | 0.0009 |
| 11 | *0* | Y = | 0.1525 | +2.160E-01 | -7.732E-02 | -9.733E-01 | 0.0009 |
| | | Z = | 0.2540 | -1.713E-02 | -9.970E-01 | +7.540E-02 | 0.0009 |
| | | X = | 0.2354 | -3.059E-01 | -2.007E-03 | +9.521E-01 | 0.0034 |
| 4 | *0* | Y = | 0.2550 | -9.471E-01 | +1.023E-01 | -3.041E-01 | 0.0033 |
| | | Z = | 0.0974 | -9.677E-02 | -9.948E-01 | -3.318E-02 | 0.0033 |
| | | X = | 0.0825 | -9.052E-01 | +1.252E-01 | -4.062E-01 | 0.0009 |
| 13 | *0* | Y = | 0.2540 | +4.031E-01 | -5.023E-02 | -9.138E-01 | 0.0009 |
| | | Z = | 0.1991 | +1.348E-01 | +9.909E-01 | +5.015E-03 | 0.0009 |
| | | X = | 0.0462 | -2.025E-01 | -1.978E-03 | +9.793E-01 | 0.0034 |
| 3 | *0* | Y = | 0.2550 | -9.671E-01 | -1.569E-01 | -2.003E-01 | 0.0033 |
| | | Z = | 0.0192 | +1.541E-01 | -9.876E-01 | +2.986E-02 | 0.0033 |

S U M M A R Y   S T A T I S T I C S   F O R   O B J E C T   P O I N T S

RMS For Standard Deviations

| | | | | |
|---|---|---|---|---|
| Count = | 7 | X = | 0.0109 | meters |
| Count = | 7 | Y = | 0.0072 | meters |
| Count = | 7 | Z = | 0.0095 | meters |

hesus X-RAY X-corner[22.5 Deg Rotation w/o T-PLATE HELD]

C O R R E C T I O N S   A P P L I E D   T O   O B J E C T   C O N T R O L

| | | | | | |
|---|---|---|---|---|---|
| 10 | X = | 0.0000 m | 1 | X = | 0.0006 m |
| | Y = | 0.0000 m | | Y = | -0.0004 m |
| | Z = | 0.0000 m | | Z = | 0.0002 m |
| 11 | X = | 0.0000 m | 2 | X = | -0.0008 m |
| | Y = | 0.0001 m | | Y = | -0.0005 m |
| | Z = | -0.0001 m | | Z = | -0.0001 m |
| 12 | X = | 0.0000 m | 3 | X = | -0.0007 m |
| | Y = | 0.0001 m | | Y = | 0.0010 m |
| | Z = | 0.0000 m | | Z = | -0.0002 m |
| 13 | X = | 0.0000 m | 4 | X = | 0.0007 m |
| | Y = | 0.0000 m | | Y = | 0.0010 m |
| | Z = | 0.0000 m | | Z = | 0.0002 m |
| 5 | X = | -0.0001 m | 6 | X = | 0.0001 m |
| | Y = | -0.0010 m | | Y = | -0.0001 m |
| | Z = | 0.0000 m | | Z = | -0.0003 m |
| 7 | X = | -0.0003 m | 8 | X = | -0.0005 m |
| | Y = | 0.0000 m | | Y = | -0.0011 m |
| | Z = | 0.0007 m | | Z = | 0.0009 m |
| 9 | X = | 0.0000 m | | | |
| | Y = | -0.0001 m | | | |
| | Z = | 0.0001 m | | | |

X .... Number of Components = 13 RMS = 0.0004 meters
Y .... Number of Components = 13 RMS = 0.0006 meters
Z .... Number of Components = 13 RMS = 0.0003 meters

A N T H R O P O M E T R Y    O U T P U T

T-PLATE ORIGIN WITH RESPECT TO HEAD ANATOMICAL ORIGIN

X= 4.6718cm    Y= -0.3749cm    Z= 4.5798cm

T-PLATE ORIENTATION WITH RESPECT TO HEAD ANATOMICAL SYSTEM

| | | |
|---|---|---|
| -0.998346 | 0.057322 | -0.004444 |
| -0.057480 | -0.996797 | 0.055606 |
| -0.001243 | 0.055769 | 0.998443 |

# Appendix 2

# PC Prep

## Sample Run

## 14 June 1990

Input data for the Preprocessing Program (PREP):

```
OPTIONS CARD:
   3,4,5,6,8 in col. 1  3,4,5,6,8-parameter transformation
   1         in col. 2  means to correct for atmospheric refraction
   1         in col. 3  means to multiply input by 25.4 (inches to mm)

CALIBRATED FIDUCIAL CARDS (one for each)FORMAT (2X,I4,4X,2F10.4)
END OF CALIBRATED FIDUCIAL MARKER:      0 in COLUMNS 1-10

   Radial Lens Distortion functions in  FORMAT (3E10.5/3E10.5)
   Decent Lens Distortion functions in  FORMAT (3E10.5)
   Atmospheric Refraction # of entries  FORMAT (I2)
   Atmospheric Refraction data in table FORMAT (2F10.3)(only if prev>0)

REPEAT FOR EACH FRAME MEASURED:
     MEASURED DATA SET:
        Frame IDentification in            FORMAT (A8)
        Observed Fiducial Coordinates in   FORMAT (6X,I4,6F10.3)
     BLANK CARD
        Observed Plate Coordinates in      FORMAT (2X,A8,6F10.3)
END OF JOB CARD:             ********** (ASTERISKS IN COLUMNS 1-10.)
```

Sample Input: (output follows)

```
301                                 Preprocessor Options: # param, atmos, inches
   111       0.0       0.0          LAT FIDUCIAL
   222      -0.018     4.728        LAT FIDUCIAL
    0

0.0       0.0       0.0             Radial  Distortion
0.0       0.0       0.0             Radial  Distortion
0.0       0.0       0.0             Tangential Distortion
 0                                  # Entries for Atmospheric Refraction
  LAT                               Frame ID
       111      5265      2102
       222      5247      6830

         5      1353      6303
        13      5247      6830
         8      9756      6349
        11      1184      2078
        10      5265      2102
         9     10084      2077
         6      1436     -2103
        12      5296     -2665
         7      9408     -2127
       lam      5989      8095
       ram      5989      8095
       lon      2930      7913
       ron      3102      7453
       ctp      3552      9811
       ltp      6421     10443
       rtp      6605      9733

**********
```

Sample Output file for the preceeding input file.

PC Giant Preprocessor June 1990

Calibrated Fiducial Coordinates

| | | |
|---|---|---|
| 111 | 0.000 | 0.000 |
| 222 | -0.457 | 120.091 |

Lens Distortion

Radial Parameters

K1= 0.00000000E+00 K2= 0.00000000E+00 K3= 0.00000000E+00
K4= 0.00000000E+00 K5= 0.00000000E+00 K6= 0.00000000E+00

(page break)

PC Giant Preprocessor June 1990

Fiducial Measurements of Frame LAT

| ID | Average | | Max Spread | |
|---|---|---|---|---|
| | X | Y | X | Y |
| 111 | 133.731 | 53.391 | 0.000 | 0.000 |
| 222 | 133.274 | 173.482 | 0.000 | 0.000 |

3-Parameter Residuals of the Fiducial Coordinates

| | | |
|---|---|---|
| 111 | 0.000 | 0.000 |
| 222 | 0.000 | 0.000 |

PLATE COORDINATES

| ID | Measured | | Adjusted | |
|---|---|---|---|---|
| | X | Y | X | Y |
| 5 | 34.366 | 160.096 | -99.365 | 106.705 |
| 13 | 133.274 | 173.482 | -0.457 | 120.091 |
| 8 | 247.802 | 161.265 | 114.071 | 107.874 |
| 11 | 30.074 | 52.781 | -103.657 | -0.610 |
| 10 | 133.731 | 53.391 | 0.000 | 0.000 |
| 9 | 256.134 | 52.756 | 122.403 | -0.635 |
| 6 | 36.474 | -53.416 | -97.257 | -106.807 |
| 12 | 134.518 | -67.691 | 0.787 | -121.082 |
| 7 | 238.963 | -54.026 | 105.232 | -107.417 |
| lam | 152.121 | 205.613 | 18.390 | 152.222 |
| ram | 152.121 | 205.613 | 18.390 | 152.222 |
| lon | 74.422 | 200.990 | -59.309 | 147.599 |
| ron | 78.791 | 189.306 | -54.940 | 135.915 |
| ctp | 90.221 | 249.199 | -43.510 | 195.809 |
| ltp | 163.093 | 265.252 | 29.362 | 211.861 |
| rtp | 167.767 | 247.218 | 34.036 | 193.827 |

# Appendix 3

# PC Giant

## Source Code

## 14 June 1990

# PC Giant

## Source Code

## File Name: 1.FOR (Input)

## 14 June 1990

```
      PROGRAM GIANT
C
C         GENERAL INTEGRATED ANALYTICAL TRIANGULATION (GIANT)
C
C  THIS IS THE MAIN CALLING PROGRAM IN THE GIANT TRIANGULATION SYSTEM.
C
      INCLUDE 'PAGEN.INC'
      INCLUDE 'TAPES.INC'
      IN=11
      IO=12
      IOS=13
      IP1=14
      IP2=15
      CAMERA=IN
      IMAGES=16
      FRAMES=IN
      OBJECT=IN
      ITAPE1=17
      ITAPE2=18
      ITAPE3=19
      ITAPE4=20
      ITAPE5=21
      ITAPE6=22
      ITAPE7=23
      ITAPE0=24
C
      OPEN (UNIT=IN,STATUS='UNKNOWN',FILE='opt.dat')
      OPEN (UNIT=IMAGES,STATUS='OLD',FILE='img.dat')
      OPEN (UNIT=IO,STATUS='UNKNOWN',FILE='giant.out',
     .                   CARRIAGE CONTROL='FORTRAN')
      OPEN (UNIT=IOS,STATUS='UNKNOWN',FILE='giant80.out',
     .                   CARRIAGE CONTROL='FORTRAN')
C
      DO 1010 I=ITAPE1,ITAPE6
           OPEN (UNIT=I,STATUS='SCRATCH',FORM='UNFORMATTED')
 1010 CONTINUE
C
C  Initialize job title, page count, and data set identifications
C
      IPAGE=-1
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
C
C  Perform data input and structuring phase, then close input files.
C
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL PHASE1
      CLOSE (IN)
      CLOSE (IMAGES)
C
C  Perform triangulation phase
```

```
C
      OPEN (UNIT=ITAPE0,STATUS='UNKNOWN')
      OPEN (UNIT=ITAPE7,STATUS='SCRATCH',FORM='UNFORMATTED')
C
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      WRITE (*,1020)
      CALL PHASE2
C
C  Perform data output phase
C
      CLOSE (ITAPE0)
      CLOSE (ITAPE1)
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      WRITE (*,1030)
      CALL PHASE3
      CALL BEEP
      CALL CLR
      CALL TOPLFT
      CALL BEEP
C
C
 1020 FORMAT (37X,'PHASE 2')
 1030 FORMAT (37X,'PHASE 3')
      END


      SUBROUTINE PHASE1
C
C  THIS is the main calling routine for
C  the data input and structuring phase
C
      INCLUDE 'TAPES.INC'
C
C  Read input data
C
      CALL RDFRAM (ITAPE3,FRAMES,OBJECT,CAMERA,IMAGES)
C
C  Organize block for autoray algorithm
C
      CALL BLOCKD (ITAPE4,ITAPE5,ITAPE3)
      CALL MERGEG (ITAPE1,ITAPE2,ITAPE3,ITAPE5,ITAPE6)
C
      RETURN
      END


      SUBROUTINE RDFRAM (ITAPE,JTAPE,KTAPE,LTAPE,MTAPE)
C
C  READ AND CODE PLATE DATA
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
      REAL*4            f,VARX,VARY,X,Y
      CHARACTER*1       LEADZ
      CHARACTER*80      INFM1,INFM2
      CHARACTER*17      IGRPH(0:1)
      CHARACTER*15      IDMSS(6,2),IDMS11,IDMS21,IDMS31,IDMS41,IDMS51,
     .                  IDMS61,IDMS12,IDMS22,IDMS32,IDMS42,IDMS52,IDMS62
      COMMON /TAPES/ IN,IO,IOS,IDUM(14)
      INCLUDE 'TITLEP.INC'
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK11.INC'
      INCLUDE 'OPTION.INC'
      INCLUDE 'OPTON2.INC'
      INCLUDE 'OPTON4.INC'
      INCLUDE 'CONVCR.INC'
      INCLUDE 'EARTHD.INC'
      INCLUDE 'PAGEN.INC'
      INCLUDE 'SWITCH.INC'
      INCLUDE 'WARNGS.INC'
C
      DIMENSION          INDXP(3,ISZ3),IDGPS(2,ISZ3),IDATA(4,100),
     .                   GP(6), FMGES(2,4), IDMGS(2,4),
     .                   IDUPL(2,200), ICODES(6),GDCOOR(6,ISZ3),
     .                   IDC12(2), IDPT12(2), ID12(2), AJPARM(2)
C
      EQUIVALENCE       (IDC1,IDC12(1)),   (IDC2,IDC12(2)),
     .                  (IDPT1,IDPT12(1)),(IDPT2,IDPT12(2)),
     .                  (ID1,ID12(1)),     (ID2,ID12(2)),
     .                  (IDMSS(1,1),IDMS11), (IDMSS(2,1),IDMS21),
     .                  (IDMSS(3,1),IDMS31), (IDMSS(4,1),IDMS41),
     .                  (IDMSS(5,1),IDMS51), (IDMSS(6,1),IDMS61),
     .                  (IDMSS(1,2),IDMS12), (IDMSS(2,2),IDMS22),
     .                  (IDMSS(3,2),IDMS32), (IDMSS(4,2),IDMS42),
     .                  (IDMSS(5,2),IDMS52), (IDMSS(6,2),IDMS62),
     .                  (X,IX), (Y,IY)
C
      DATA INFM1        /'(2A4,3F12.3,3F10.3)'/
      DATA INFM2        /'(2A4,3F12.3,3F10.3,5X,I1)'/
      DATA IGRPH        /'(Photo to Object)','(Object to Photo)'/
      DATA IEND         /'****'/
      DATA NMAX         /ISZ1/
      DATA MMAX         /ISZ2/
      DATA LMAX         /ISZ3/
      DATA MAXD         /200/
      DATA ICODES       /1,1,0,1,1,0/
      DATA ZERO         /0.0D0/
      DATA MAXLIN       /57/
C
C  Initialization
C
      IS=0
      IDCAM(1,1)=IEND
      IDCAM(2,1)=IEND
      IDPLT(1,1)=IEND
      IDPLT(2,1)=IEND
      N=0
```

```
      M=0
      DO 1010 I=1,NMAX
           INDEX(1,I)=0
           INDEX(2,I)=I
 1010 CONTINUE
C
C  Rewind data sets
C
C  ** ITAPE ** Output tape for triangulation input data
C  ** JTAPE ** Input camera station parameters
C  ** KTAPE ** Input object control
C  ** LTAPE ** Input camera system parameters
C  ** MTAPE ** Input image data
C
      INF1=JTAPE
      INF2=KTAPE
      REWIND ITAPE
C
C                        GIANT PROGRAM OPTIONS:
C
C
C   cc:         OPTION:                                Variable: Format:
C
C    1   Definition of Object Space Units                 IUNIT    I1
C        = 0,  Rectangular Coordinates (Meters)
C        = 1,  Geographic Coordinates
C                    (Deg., Min., Sec., Meters)
C
C    2   Type of Camera Station Attitude Switch           IATT     I1
C        (Affecting both Input and Output)
C        = 0,  Photo to Ground
C        = 1,  Ground to Photo
C
C    3   List Input Camera Station Parameters Switch      IPSTA    I1
C        = 0,  list
C        = 1,  do not list
C
C    4   List Input Plate Coordinates Switch              IPIMG    I1
C        = 0,  list
C        = 1,  do not list
C
C    5   List Input Object Space Control                  IPCRL    I1
C        = 0,  list
C        = 1,  do not list
C
C    6   List Output Triangulated Object Point            ILTGP    I1
C        Coordinates Switch
C        = 0,  list
C        = 1,  do not list
C
C    7   Save (as a FILE) Output Triangulated             IPNGP    I1
C        Object Coordinates Switch
C        = 0,  save
C        = 1,  do not save
C
```

```
C      8   List Output Adjusted Camera Station         ILTST    I1
C          Parameters Switch
C          = 0,   list
C          = 1,   do not list
C
C      9   Save (as a FILE) Adjusted Camera Station    IPNST    I1
C          Parameters Switch
C          = 0,   save
C          = 1,   do not save
C
C     10   Triangulation Process Selection Switch      ITRNG    I1
C          = 0,   Perform COMPLETE TRIANGULATION.
C          = 1,   Perform INTERSECTION ONLY, holding
C                 Camera Positions and Attitudes fixed.
C
C     11   Error Propagation Switch for the GDOP       IPROP    I1
C           (Geometric Dilution Of Precision)
C          = 0,   do not perform Error Propagation
C          = 1,   perform Error Propagation
C
C        <<See Option "20" for type of GDOP Output.>>
C
C
C     12   "a posteriori" Unit Variance Adjustment Flag  IWGHT  I1
C          = 0,   Unit Variance  is based  on completely
C                 Free Camera Parameters.
C          = 1,   Unit Variance is based on Constrained
C                 Camera Parameters.
C          = 2,   Force Unit Variance to Unity
C                 (For Simulation Purposes).
C
C     13   Sort Triangulated Object Space Points Switch  ISORT  I1
C          = 0,   perform ascending sort of Object Points
C          = 1,   do not perform sort
C
C     14   Maximum number of Iterations allowed in the   NIT    I1
C          Least Squares Adjustment.  If this field is
C          left blank, the Default Max is 4.
C
C     15   Any valid Alphanumeric character.  Leading    LEADZ  A1
C          character(s)  which matches this character
C          will be removed from Name Fields of Camera
C          Systems, Camera Stations and Object Points.
C
C     16   Air Refraction Model Switch                 IAREFR   I1
C          = 0,   do not apply
C          = 1,   apply
C
C     17   Water Refraction Model Switch               IWREFR   I1
C          = 0,   do not apply
C          = 1,   apply
C
C  18-19   Criterion E for convergence of least squares  I      I2
C          adjustment.  Least Squares  solution will be
C          considered complete  if the absolute  change
```

```
C         in  the  weighted sum of the squares for two
C         consecutive iterations is less than E percent.
C         If this field is left blank, the program will
C         assume E = 5%.
C
C   20    Eigenvalue/vector - Variance/Covariance        IEIGEN    I1
C         = 0, all positional error  will be expressed
C              as Error Ellipsoid Orientation & Length
C                       (Eigenvectors & Eigenvalues in
C                           descending order of size.)
C              Orientation error will  be expressed as
C              Standard Deviations in Degrees Min Secs.
C         = 1, all error will be expressed as
C              Variance - Covariance Matrices with the
C              Object Space Points  also  showing  the
C              Square Roots of the Diagonal terms under
C              the heading "Standard Deviation".
C
C   21    Anthropometry Option (1 if yes)                IANTH     I1
C
C 31-40   Water level  (meters) with  respect to the     WLEVEL  F10.3
C         reference ellipsoid  at  the  time  of the
C         exposure.  This value applies to the whole
C         block for bathymetric mapping applications.
C
C
C 41-50   Plate residual listing criterion  (in  mm.)    RESIDA  F10.3
C         = 0, ALL  image  residuals  will be  listed
C         > 0, only  those residuals  whose  absolute
C              value is  greater  than the  criterion
C              will be listed.
C         < 0, No residuals will be listed.
C
C 51-60   Semimajor Axis of the Ellipsoid in Meters.     SPHRD(1)F10.2
C         If  not specified,  program will  assume  the
C         value  of the  GRS 1980  Ellipsoid (NAD 1983)
C
C 61-70   Semiminor Axis of the Ellipsoid in Meters.     SPHRD(2)F10.2
C         If  not specified,  program will  assume  the
C         value  of the  GRS 1980  Ellipsoid (NAD 1983)
C
C
C  READ TITLE CARD:
C
      READ (IN,1440) JTITLE
C
C  READ OPTIONS CARD:
C
      READ (IN,1450) IUNIT ,IATT  ,IPSTA ,IPIMG,IPCRL ,ILTGP,IPNGP,
     .               ILTST ,IPNST ,ITRNG ,IPROP,IWGHT ,ISORT,NIT   ,
     .               LEADZ ,IAREFR,IWREFR,I    ,IEIGEN,IANTH,
     .               WLEVEL,RESIDA,    SPHRD(1),SPHRD(2)
      IRESA=1000.0*RESIDA
      EPSLN=I/100.0D0
      READ (IN,1460) AJPARM
```

```
      CNW=1.8D0
      IF (AJPARM(1).LE.ZERO) AJPARM(1)=0.001D0
      IF (AJPARM(2).LE.ZERO) AJPARM(2)=0.01D0
      DVPA=1000.0D0
      DVA=900000.0D0
      DVPL=60000.0D0
      IF (ITRNG.NE.0) THEN
           IAREFR=1
           IWREFR=1
           IPROP=0
      END IF
      IF (NIT.LE.0) NIT=4
      IF (EPSLN.LE.ZERO) EPSLN=0.05D0
C
C  Default to GRS 1980 Ellipsoid of revolution (NAD 1983)
C
      IF (SPHRD(1).LE.ZERO) SPHRD(1)=6378137.D0
      IF (SPHRD(2).LE.ZERO) SPHRD(2)=6356752.3141D0
      CALL NEWPAG
      CALL LISTTP (LEADZ)
C
C  Read camera data
C
      CALL READIM (NFRM,LEADZ,LTAPE,MTAPE)
      CALL TSTFRM (INFM1,INF1,IND)
      IF (IND.EQ.0) GO TO 1270
 1020 READ (INF1,INFM1) IDC1,IDC2,GP
      IF (IDC1.EQ.IEND) GO TO 1270
      CALL REFRM (IDC12,LEADZ)
      CALL GETFR (IDC12,F,VARX,VARY)
C
C  List frame identification, principal distance, standard deviation of
C  plate-x, and standard deviation of plate-y.
C
      IF (VARX.LE.ZERO) VARX=0.01
      IF (VARY.LE.ZERO) VARY=0.01
      IF (IPSTA.NE.0.AND.IPIMG.NE.0) GO TO 1030
      CALL NEWPAG
      WRITE (IO,1470) IDC1,IDC2,F,VARX,VARY
      WRITE (IOS,1690) IDC1,IDC2,F,VARX,VARY
C
C  Code camera ID
C
 1030 DO 1040 IDC=1,N
           IF (IDC1.NE.IDCAM(1,IDC)) GO TO 1040
           IF (IDC2.EQ.IDCAM(2,IDC)) GO TO 1060
 1040 CONTINUE
      N=N+1
      IF (N.LE.NMAX) GO TO 1050
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1480) N,NMAX
      STOP
```

```
 1050 IDC=N
      IDCAM(1,IDC)=IDC1
      IDCAM(2,IDC)=IDC2
C
C  Read rest of camera parameters and store them
C
 1060 IF (IUNIT.EQ.0) GO TO 1080
      IF (GP(4).LE.ZERO) GP(4)=DVPA
      IF (GP(5).LE.ZERO) GP(5)=DVPA
      DO 1070 I=1,6
         IF (ICODES(I).EQ.0) GO TO 1070
         GP(I)=DEGRAD(GP(I))
         CALL RADDEG (GP(I),IDMSS(I,1))
 1070 CONTINUE
 1080 IF (GP(4).LE.ZERO) GP(4)=DVPL
      IF (GP(5).LE.ZERO) GP(5)=DVPL
      IF (GP(6).LE.ZERO) GP(6)=DVPL
      DO 1090 I=1,3
         J=I+3
         PARAM(I,IDC)=GP(I)
         WTMAT(I,IDC)=GP(J)
 1090 CONTINUE
      READ (INF1,INFM1) ID1,ID2,GP
      IF (GP(4).LE.ZERO) GP(4)=DVA
      IF (GP(5).LE.ZERO) GP(5)=DVA
      IF (GP(6).LE.ZERO) GP(6)=DVA
      DO 1100 I=1,6
         GP(I)=DEGRAD(GP(I))
         CALL RADDEG (GP(I),IDMSS(I,2))
 1100 CONTINUE
      DO 1110 I=1,3
         J=I+3
         PARAM(J,IDC)=GP(I)
         WTMAT(J,IDC)=GP(J)
 1110 CONTINUE
      FOCAL(IDC)=F
      VARPLT(1,IDC)=1.0/VARX
      VARPLT(2,IDC)=1.0/VARY
C
C  List camera station position and attitude
C
      CALL REFRM (ID12,LEADZ)
      IF (ID1.EQ.IDC1.AND.ID2.EQ.IDC2) GO TO 1120
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1490) IDC12,ID12
      STOP
 1120 IF (IPSTA.NE.0) GO TO 1150
 1130 WRITE (IO,1500) IGRPH(IATT)
      WRITE (IOS,1700) IGRPH(IATT)
      IF (IUNIT.NE.0) GO TO 1140
      WRITE (IO,1510) PARAM(1,IDC),WTMAT(1,IDC),IDMS12,IDMS42,PARAM(2,
     .IDC),WTMAT(2,IDC),IDMS22,IDMS52,PARAM(3,IDC),WTMAT(3,IDC),IDMS32,
```

```
     .IDMS62
      WRITE (IOS,1710) PARAM(1,IDC),WTMAT(1,IDC),IDMS12,IDMS42,PARAM(2,
     .IDC),WTMAT(2,IDC),IDMS22,IDMS52,PARAM(3,IDC),WTMAT(3,IDC),IDMS32,
     .IDMS62
      GO TO 1150
 1140 WRITE (IO,1520) IDMS11,IDMS41,IDMS12,IDMS42,IDMS21,IDMS51,IDMS22,
     .IDMS52,PARAM(3,IDC),WTMAT(3,IDC),IDMS32,IDMS62
      WRITE (IOS,1720) IDMS11,IDMS41,IDMS12,IDMS42,IDMS21,IDMS51,IDMS22,
     .IDMS52,PARAM(3,IDC),WTMAT(3,IDC),IDMS32,IDMS62
C
C  Convert Standard Deviations of position and attitude to weights
C
 1150 DO 1160 I=1,6
 1160      WTMAT(I,IDC)=1.0/WTMAT(I,IDC)**2
C
C  List title for plate coordinates
C
      IF (IPIMG.NE.0) GO TO 1170
      WRITE (IO,1530)
      WRITE (IOS,1730)
      LINES=16
      IF (IPSTA.NE.0) LINES=7
C
C  Read plate coordinate data
C
 1170 II=0
C
C  Define and position image coordinate data set
C
 1180 K=0
 1190 CALL GETPT (IDPT12,X,Y)
      IF (IDPT1.EQ.IEND.OR.IDPT2.EQ.IEND) GO TO 1250
C
C  List plate coordinates
C
      IF (IPIMG.NE.0) GO TO 1210
      II=II+1
      IDMGS(1,II)=IDPT1
      IDMGS(2,II)=IDPT2
      FMGES(1,II)=X
      FMGES(2,II)=Y
      IF (II.NE.4) GO TO 1210
      II=0
      LINES=LINES+1
      IF (LINES.LE.MAXLIN) GO TO 1200
      CALL NEWPAG
      WRITE (IO,1540) IDC1,IDC2
      WRITE (IO,1530)
      WRITE (IOS,1740) IDC1,IDC2
      WRITE (IOS,1730)
      LINES=7
 1200 WRITE (IO,1550) (IDMGS(1,I),IDMGS(2,I),FMGES(1,I),FMGES(2,I),I=1,
     .4)
      WRITE (IOS,1750) (IDMGS(1,I),IDMGS(2,I),FMGES(1,I),FMGES(2,I),I=1,
     .4)
```

```
C
C  Check to insert plate coord ident in table
C
 1210 K=K+1
      DO 1220 IDPT=1,M
           IF (IDPT1.NE.IDPLT(1,IDPT)) GO TO 1220
           IF (IDPT2.EQ.IDPLT(2,IDPT)) GO TO 1240
 1220 CONTINUE
      M=M+1
      IF (M.LE.MMAX) GO TO 1230
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1560) M,MMAX
      STOP
 1230 IDPT=M
      IDPLT(1,IDPT)=IDPT1
      IDPLT(2,IDPT)=IDPT2
C
C  Store point data
C
 1240 IF (INDEX(1,IDC).LT.IDPT) INDEX(1,IDC)=IDPT
      IDATA(1,K)=IDPT
      IDATA(2,K)=IX
      IDATA(3,K)=IY
      IDATA(4,K)=IDC
      IF (K.NE.100) GO TO 1190
      WRITE (ITAPE) K,((IDATA(I,J),I=1,4),J=1,K)
      GO TO 1180
C
C  End of plate data
C
 1250 IF (IPIMG.NE.0.OR.II.EQ.0) GO TO 1260
      WRITE (IO,1550) (IDMGS(1,I),IDMGS(2,I),FMGES(1,I),FMGES(2,I),I=1,
     .II)
      WRITE (IOS,1750) (IDMGS(1,I),IDMGS(2,I),FMGES(1,I),FMGES(2,I),I=1,
     .II)
 1260 IF (K.NE.0) WRITE (ITAPE) K,((IDATA(I,J),I=1,4),J=1,K)
      GO TO 1020
C
C  Write images sentinel,
C  if geographic,  compute mean latitude and longitude
C  write camera station data
C
 1270 K=1
      IDATA(1,1)=0
      WRITE (ITAPE) K,(IDATA(I,1),I=1,4)
      WRITE (ITAPE) N,((PARAM(I,J),I=1,6),J=1,N),((VARPLT(I,J),I=1,2),J=
     .1,N),(FOCAL(I),I=1,N),((WTMAT(I,J),I=1,6),J=1,N),((IDCAM(I,J),I=1,
     .2),J=1,N)
      WRITE (ITAPE) M,((IDPLT(I,J),I=1,2),J=1,M)
C
C  Initialize for object space control data
C
```

```
      NG=0
      NPTP=10
      NPTF=0
      INPCTR=0
      NDUPL=0
      NCNTRL=0
C
C  Read object space control points:
C
      CALL TSTFRM (INFM2,INF2,IND)
C
C  Test to see if any control exists; if none then get out & write
C  flag (NCNTRL=1) for appropriate action when printing output report
C  such that the "CORRECTIONS TO OBJECT SPACE CONTROL" (Last Page) is
C  not computed or printed.
C
      IF (IND.EQ.0) THEN
           NCNTRL=1
      END IF
 1280 READ (INF2,INFM2) ID1,ID2,GP,IND
      IF (ID1.EQ.'****') GO TO 1410
      CALL REFRM (ID12,LEADZ)
      IF (IND.LT.0.OR.IND.GT.7) IND=7
      IF (NG.LT.LMAX) GO TO 1290
C
C  Number of Ground (NG) control points just read exceeds LMAX.
C  \              Write Error Message & STOP.
C
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1570) NG,LMAX
      STOP
C
C  List the object space control points, if any
C
 1290 IF (IUNIT.EQ.0) GO TO 1310
      IF (GP(4).LE.ZERO) GP(4)=AJPARM(1)
      IF (GP(5).LE.ZERO) GP(5)=AJPARM(1)
      DO 1300 I=1,6
           IF (ICODES(I).EQ.0) GO TO 1300
           GP(I)=DEGRAD(GP(I))
           CALL RADDEG (GP(I),IDMSS(I,1))
 1300 CONTINUE
      GO TO 1320
 1310 IF (GP(4).LE.ZERO) GP(4)=AJPARM(1)
      IF (GP(5).LE.ZERO) GP(5)=AJPARM(1)
 1320 IF (GP(6).LE.ZERO) GP(6)=AJPARM(2)
      IF (IPCRL.NE.0) GO TO 1350
      IF (NPTF.NE.0) GO TO 1330
C
C  Call for new page & print title for list of object space control
C
      CALL NEWPAG
```

```
      WRITE (IO,1580)
      WRITE (IOS,1760)
 1330 NPTF=NPTF+1
      IF (NPTF.EQ.NPTP) NPTF=0
      IF (IUNIT.NE.0) GO TO 1340
      WRITE (IO,1590) GP(1),GP(4)
      WRITE (IO,1600) ID1,ID2,GP(2),GP(5),IND
      WRITE (IO,1610) GP(3),GP(6)
      WRITE (IOS,1770) GP(1),GP(4)
      WRITE (IOS,1780) ID1,ID2,GP(2),GP(5),IND
      WRITE (IOS,1790) GP(3),GP(6)
      GO TO 1350
 1340 WRITE (IO,1620) IDMS11,IDMS41
      WRITE (IO,1630) ID1,ID2,IDMS21,IDMS51,IND
      WRITE (IO,1640) GP(3),GP(6)
      WRITE (IOS,1800) IDMS11,IDMS41
      WRITE (IOS,1810) ID1,ID2,IDMS21,IDMS51,IND
      WRITE (IOS,1820) GP(3),GP(6)
C
C  Convert standard deviations to weights
C
 1350 GP(4)=1.0D0/GP(4)**2
      GP(5)=1.0D0/GP(5)**2
      GP(6)=1.0D0/GP(6)**2
C
C  Check if point is photographed
C
      DO 1360 I=1,M
           IF (ID1.NE.IDPLT(1,I)) GO TO 1360
           IF (ID2.EQ.IDPLT(2,I)) GO TO 1370
 1360 CONTINUE
      INPCTR=INPCTR+1
      IDGPS(1,INPCTR)=ID1
      IDGPS(2,INPCTR)=ID2
      GO TO 1280
 1370 IF (NG.EQ.0) GO TO 1400
      DO 1380 J=1,NG
           K=INDXP(1,J)
           IF (K.EQ.I) GO TO 1390
 1380 CONTINUE
      GO TO 1400
 1390 IF (NDUPL.EQ.MAXD) GO TO 1280
      NDUPL=NDUPL+1
      IDUPL(1,NDUPL)=ID1
      IDUPL(2,NDUPL)=ID2
      GO TO 1280
 1400 NG=NG+1
      INDXP(1,NG)=I
      INDXP(2,NG)=NG
      INDXP(3,NG)=IND
      DO 1405, IX =1, 6
          GDCOOR(IX,NG)=GP(IX)
 1405 CONTINUE
      GO TO 1280
C
```

```
C  Write object space control data and list unphotographed points
C
 1410 WRITE (ITAPE) N,((INDEX(I,J),I=1,2),J=1,N)
      WRITE (ITAPE) NG,((INDXP(I,J),I=1,3),J=1,NG),((GDCOOR(I,J),I=1,6),
     .J=1,NG)
      IF (INPCTR.EQ.0.AND.NDUPL.EQ.0) GO TO 1430
      CALL NEWPAG
      WRITE (IO,1650)
      WRITE (IOS,1830)
      IF (INPCTR.EQ.0) GO TO 1420
      WRITE (IO,1660)
      WRITE (IO,1670) ((IDGPS(I,J),I=1,2),J=1,INPCTR)
      WRITE (IOS,1840)
      WRITE (IOS,1850) ((IDGPS(I,J),I=1,2),J=1,INPCTR)
      IF (NDUPL.EQ.0) GO TO 1430
 1420 WRITE (IO,1680)
      WRITE (IO,1670) ((IDUPL(I,J),I=1,2),J=1,NDUPL)
      WRITE (IOS,1860)
      WRITE (IOS,1850) ((IDUPL(I,J),I=1,2),J=1,NDUPL)
C
 1430 RETURN
C
C
 1440 FORMAT (20A4)
 1450 FORMAT (14I1,A1,2I1,I2,2I1,9X,2F10.3,2F10.2)
 1460 FORMAT (2F10.3)
 1470 FORMAT (54X,'FRAME ',2A4//11X,'PRINCIPAL DISTANCE =',F10.4,' mm',
     .6X,'ST. D. OF X = ',F6.4,' mm',7X,'ST. D. OF Y = ',F6.4,' mm'//)
 1480 FORMAT (20X,I4,' CAMERA STATIONS EXCEEDED ',I4)
 1490 FORMAT (' CAMERA POSITION ID ',2A4,' DOES NOT MATCH CAMERA ATTITUD
     .E ID ',2A4)
 1500 FORMAT (47X,'CAMERA STATION PARAMETERS',/23X,'P O S I T I O N',
     .38X,'A T T I T U D E ',A17//)
 1510 FORMAT (11X,'X = ',F11.4,' m',3X,'ST. D. = ',F11.4,' m',10X,'OMEGA
     . = ',A15,5X,'ST. D. = ',A15/11X,'Y = ',F11.4,' m',3X,'ST. D. = ',
     .F11.4,' m',10X,'PHI   = ',A15,5X,'ST. D. = ',A15/11X,'Z = ',F11.4,
     .' m',3X,'ST. D. = ',F11.4,' m',10X,'KAPPA = ',A15,5X,'ST. D. = ',
     .A15//)
 1520 FORMAT (6X,'LNG = ',A15,3X,'ST. D. = ',A15,10X,'OMEGA = ',A15,5X,'
     .ST. D. = ',A15/6X,'LAT = ',A15,3X,'ST. D. = ',A15,10X,'PHI   = ',
     .A15,5X,'ST. D. = ',A15/6X,'ELV = ',F15.4,3X,'ST. D. = ',F15.4,10X,
     .'KAPPA = ',A15,5X,'ST. D. = ',A15//)
 1530 FORMAT (45X,'PLATE COORDINATES in millimeters',/7X,'ID',7X,'X',9X,
     .'Y',3(12X,'ID',7X,'X',9X,'Y')/)
 1540 FORMAT (54X,'FRAME ',2A4//)
 1550 FORMAT (1X,2A4,2F10.4,3(4X,2A4,2F10.4))
 1560 FORMAT (20X,I5,' IMAGE POINTS EXCEEDED ',I5)
 1570 FORMAT (20X,I4,' OBJECT CONTROL EXCEEDED ',I4)
 1580 FORMAT (47X,'O B J E C T   C O N T R O L   D A T A'////)
 1590 FORMAT (45X,'X = ',F11.4,' m',5X,'ST. D. = ',F9.4)
 1600 FORMAT (31X,2A4,6X,'Y = ',F11.4,' m',5X,'ST. D. = ',F9.4,5X,'TYPE
     .= ',I1)
 1610 FORMAT (45X,4HZ = ,F11.4,' m',5X,9HST. D. = ,F9.4//)
 1620 FORMAT (42X,'LNG = ',A15,5X,'ST. D. = ',A15)
 1630 FORMAT (28X,2A4,6X,'LAT = ',A15,5X,'ST. D. = ',A15,4X,'TYPE = ',
```

```
     .I1)
 1640 FORMAT (42X,'ELV = ',F15.4,5X,'ST. D. = ',F15.4//)
 1650 FORMAT (52X,'E R R O R    W A R N I N G S'////)
 1660 FORMAT (54X,'POINTS NOT PHOTOGRAPHED',/)
 1670 FORMAT (44X,2A4,4X,2A4,4X,2A4,4X,2A4)
 1680 FORMAT (//54X,'DUPLICATE CONTROL POINTS'/)
C  80 col
 1690 FORMAT (//32X,'FRAME ',2A4//22X,'PRINCIPAL DISTANCE =',F10.4,' mm'
     ./25X,'Std. Dev.. of X = ',F6.4,' mm'/25X,'Std. Dev. of Y = ',F6.4,'
     . mm'//)
 1700 FORMAT (25X,'CAMERA STATION PARAMETERS'//4X,'P O S I T I O N',8X,'
     .Std. Dev.',8X,'A T T I T U D E',8X,'Std. Dev.'/43X,A17/)
 1710 FORMAT ('    X = ',F11.4,' m',4X,F11.4,' m',3X,'OMEGA = ',2A15/
     .          '    Y = ',F11.4,' m',4X,F11.4,' m',3X,'PHI   = ',2A15/
     .          '    Z = ',F11.4,' m',4X,F11.4,' m',3X,'KAPPA = ',2A15//)
 1720 FORMAT (' LNG = ',2A15,2X,'OMEGA = ',2A15/' LAT = ',2A15,2X,'PHI
     . = ',2A15/' ELV = ',2F15.4,2X,'KAPPA = ',2A15///)
 1730 FORMAT (24X,'PLATE COORDINATES in millimeters',/2(11X,'ID',7X,'X',
     .8X,'Y',4X)/)
 1740 FORMAT (32X,'FRAME ',2A4//)
 1750 FORMAT (2(6X,2A4,2F10.4,2X))
 1760 FORMAT (20X,'O B J E C T    C O N T R O L    D A T A'//25X,'Positi
     .on                          Std. Dev.'//)
 1770 FORMAT (21X,'X = ',F11.4,' m',5X,F9.4)
 1780 FORMAT (7X,2A4,6X,'Y = ',F11.4,' m',5X,F9.4,5X,'TYPE = ',I1)
 1790 FORMAT (21X,'Z = ',F11.4,' m',5X,F9.4//)
 1800 FORMAT (20X,'LNG = ',A15,5X,A15)
 1810 FORMAT (7X,2A4,5X,'LAT = ',2(A15,5X),'TYPE = ',I1)
 1820 FORMAT (20X,'ELV = ',F15.4,5X,F15.4//)
 1830 FORMAT (27X,'E R R O R    W A R N I N G S'////)
 1840 FORMAT (29X,'POINTS NOT PHOTOGRAPHED'/)
 1850 FORMAT ((15X,4(4X,2A4)))
 1860 FORMAT (//29X,'DUPLICATE CONTROL POINTS'/)
      END


      SUBROUTINE NEWPAG
C
C   GENERATE TITLE PAGES FOR GIANT SYSTEM.
C
      INCLUDE 'TAPES.INC'
      INCLUDE 'TITLEP.INC'
      INCLUDE 'PAGEN.INC'
C
      IPAGE=IPAGE+1
      IF (IPAGE .GT. 0) THEN
          WRITE (IO,1010) JTITLE,IPAGE
          WRITE (IOS,1020) IPAGE,JTITLE
      ENDIF
C
      RETURN
C
C
 1010 FORMAT('1MS-DOS/VMS/UNIX GIANT (5/90) :',3X,20A4,3X,'PAGE',I5//)
 1020 FORMAT('1MS-DOS/VMS/UNIX GIANT (5/90) :',38X,'PAGE',I5/1X,20A4/)
```

```
      END


      SUBROUTINE LISTTP (LEADZ)
C
C  The purpose of this routine is to list various GIANT parameters
C
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
      CHARACTER*1 LEADZ
      INCLUDE 'OPTION.INC'
      INCLUDE 'OPTON2.INC'
      INCLUDE 'OPTON4.INC'
      INCLUDE 'CONVCR.INC'
      INCLUDE 'EARTHD.INC'
      INCLUDE 'TAPES.INC'
      RESIDA=IRESA/1000.
      CALL CLR
      CALL TOPLFT
C
      IF (IUNIT.EQ.0) THEN
           WRITE (*,1290)
           WRITE (IO,1010)
           WRITE (IOS,1290)
      ELSE
           WRITE (*,1300)
           WRITE (IO,1020)
           WRITE (IOS,1300)
      END IF
C
      IF (IATT.EQ.0) THEN
           WRITE (*,1310)
           WRITE (IO,1030)
           WRITE (IOS,1310)
      ELSE
           WRITE (*,1320)
           WRITE (IO,1040)
           WRITE (IOS,1320)
      END IF
C
      IF (ITRNG.EQ.0) THEN
           WRITE (*,1330)
           WRITE (IO,1050)
           WRITE (IOS,1330)
C  If Error Propagation is desired, then:
           IF (IPROP.EQ.1) THEN
                WRITE (*,1340)
                WRITE (IO,1060)
                WRITE (IOS,1340)
C  If Eigenvector / Eigenvalue output is desired,  then:
                IF (IEIGEN.EQ.0) THEN
                     WRITE (*,1350)
                     WRITE (IO,1070)
                     WRITE (IOS,1350)
C  Else Variance / Covariance output is desired:
                ELSE
```

```
                  WRITE (*,1360)
                  WRITE (IO,1080)
                  WRITE (IOS,1360)
             END IF
C  If Unit Variance is based on Completely free Cameras, then:
             IF (IWGHT.EQ.0) THEN
                  WRITE (*,1370)
                  WRITE (IO,1090)
                  WRITE (IOS,1370)
C  Else If Unit Variance is based on Constrained Cameras, then:
             ELSE IF (IWGHT.EQ.1) THEN
                  WRITE (*,1380)
                  WRITE (IO,1100)
                  WRITE (IOS,1380)
C  Else Unit Variance is being FORCED to Unity (for Project Design):
             ELSE
                  WRITE (*,1390)
                  WRITE (IO,1110)
                  WRITE (IOS,1390)
             END IF
        ELSE
C  Else Error Propagation is not desired.
             WRITE (*,1400)
             WRITE (IO,1120)
             WRITE (IOS,1400)
        END IF
C  \
        IF (IUNIT.NE.0) THEN
             IF (IAREFR.EQ.0) THEN
                  WRITE (IO,1130)
                  WRITE (IOS,1130)
             ELSE
                  WRITE (IO,1140)
             END IF
             IF (IWREFR.EQ.0) THEN
                  WRITE (IO,1150)
                  WRITE (IO,1160) WLEVEL
             ELSE
                  WRITE (IO,1170)
             END IF
        END IF
     ELSE
        WRITE (*,1410)
        WRITE (IO,1180)
        WRITE (IOS,1410)
     END IF
C
     IF (RESIDA.EQ.0.0) THEN
        WRITE (*,1420)
        WRITE (IO,1190)
        WRITE (IOS,1420)
     ELSE IF (RESIDA.GT.0.0) THEN
        WRITE (*,1430) RESIDA
        WRITE (IO,1200) RESIDA
        WRITE (IOS,1430) RESIDA
```

```
      ELSE
           WRITE (*,1440)
           WRITE (IO,1210)
           WRITE (IOS,1440)
      END IF
C
      WRITE (IO,1220) LEADZ
C
      IF (IUNIT.NE.0) THEN
           WRITE (*,1450) SPHRD(1)
           WRITE (IO,1230) SPHRD(1)
           WRITE (IOS,1450) SPHRD(1)
           WRITE (IO,1240) SPHRD(2)
      END IF
      IF (IPNGP.EQ.0) THEN
           WRITE (*,1460)
           WRITE (IO,1250)
           WRITE (IOS,1460)
      ELSE
           WRITE (*,1470)
           WRITE (IO,1260)
           WRITE (IOS,1470)
      END IF
C
      IF (IPNST.EQ.0) THEN
           WRITE (*,1480)
           WRITE (IO,1270)
           WRITE (IOS,1480)
      ELSE
           WRITE (*,1490)
           WRITE (IO,1280)
           WRITE (IOS,1490)
      END IF
C
      RETURN
C
C  The following are messages to 132 column hardcopy:
C
 1010 FORMAT (10(/),43X,'OBJECT SPACE REFERENCE SYSTEM IS RECTANGULAR')
 1020 FORMAT (10(/),45X,'OBJECT SPACE REFERENCE SYSTEM IS GEOGRAPHIC')
 1030 FORMAT (/,49X,'ROTATION ANGLES ARE PHOTO-TO-OBJECT')
 1040 FORMAT (/,49X,'ROTATION ANGLES ARE OBJECT-TO-PHOTO')
 1050 FORMAT (/,45X,'COMPLETE TRIANGULATION PROCESS IS REQUESTED')
 1060 FORMAT (/,51X,'ERROR PROPAGATION IS REQUESTED')
 1070 FORMAT (/,51X,'[EIGENVECTOR/EIGENVALUE OUTPUT]')
 1080 FORMAT (/,53X,'[VARIANCE/COVARIANCE OUTPUT]')
 1090 FORMAT (/,34X,'UNIT VARIANCE WILL BE BASED ON COMPLETELY FREE CAME
     .RA PARAMETERS')
 1100 FORMAT (/,36X,'UNIT VARIANCE WILL BE BASED ON CONSTRAINED CAMERA P
     .ARAMETERS')
 1110 FORMAT (/,48X,'UNIT VARIANCE WILL BE FORCED TO UNITY')
 1120 FORMAT (/,49X,'ERROR PROPAGATION IS NOT REQUESTED')
 1130 FORMAT (/,38X,'ATMOSPHERIC REFRACTION WILL BE INCLUDED IN THE ADJU
     .STMENT')
 1140 FORMAT (/,36X,'ATMOSPHERIC REFRACTION WILL NOT BE INCLUDED IN THE
```

```
     .ADJUSTMENT')
 1150 FORMAT (/,41X,'WATER REFRACTION WILL BE INCLUDED IN THE ADJUSTMENT
     .')
 1160 FORMAT (/,41X,'WATER LEVEL AT TIME OF PHOTOGRAPHY =',F7.3,' METERS
     .')
 1170 FORMAT (/,39X,'WATER REFRACTION WILL NOT BE INCLUDED IN THE ADJUST
     .MENT')
 1180 FORMAT (/,50X,'INTERSECTION PROCESS IS REQUESTED')
 1190 FORMAT (/,49X,'ALL IMAGE RESIDUALS WILL BE LISTED')
 1200 FORMAT (/,39X,'IMAGE RESIDUALS GREATER THAN',F7.3,' (mm) WILL BE L
     .ISTED')
 1210 FORMAT (/,50X,'NO IMAGE RESIDUAL WILL BE LISTED')
 1220 FORMAT (/,38X,'LEADING ''',A1,''' WILL BE ELIMINATED FROM ALL IDEN
     .TIFICATIONS')
 1230 FORMAT (/,40X,'Semi-Major axis of ELLIPSOID (a) = ',F11.3,' meters
     .')
 1240 FORMAT (/,40X,'Semi-Minor axis of ELLIPSOID (b) = ',F11.3,' meters
     .')
 1250 FORMAT (/,44X,'TRIANGULATED OBJECT COORDINATES WILL BE SAVED')
 1260 FORMAT (/,42X,'TRIANGULATED OBJECT COORDINATES WILL NOT BE SAVED')
 1270 FORMAT (/,42X,'ADJUSTED CAMERA STATION PARAMETERS WILL BE SAVED')
 1280 FORMAT (/,40X,'ADJUSTED CAMERA STATION PARAMETERS WILL NOT BE SAVE
     .D')
C
C  The following are messages to the screen and 80 column hardcopy:
C
 1290 FORMAT (/,18X,'Object Space Reference System is Rectangular')
 1300 FORMAT (/,19X,'Object Space Reference System is Geographic')
 1310 FORMAT (/,23X,'Rotation angles are Photo-to-Object')
 1320 FORMAT (/,23X,'Rotation Angles are Object-to-Photo')
 1330 FORMAT (/,19X,'Complete Triangulation process is requested')
 1340 FORMAT (/,25X,'Error Propagation is requested')
 1350 FORMAT (/,25X,'[Eigenvector/Eigenvalue output]')
 1360 FORMAT (/,27X,'[Variance/Covariance output]')
 1370 FORMAT (/,8X,'Unit Variance will be based on completely free camer
     .a parameters')
 1380 FORMAT (/,10X,'Unit Variance will be based on constrained camera p
     .arameters')
 1390 FORMAT (/,22X,'Unit Variance will be forced to unity')
 1400 FORMAT (/,23X,'Error Propagation is not requested')
 1410 FORMAT (/,24X,'INTERSECTION PROCESS IS REQUESTED')
 1420 FORMAT (/,23X,'All Image Residuals will be listed')
 1430 FORMAT (/,13X,'Image Residuals greater than',F7.3,' (mm) will be l
     .isted')
 1440 FORMAT (/,24X,'No Image Residual will be listed')
 1450 FORMAT (/,14X,'Semi-Major axis of ELLIPSOID (a) = ',F11.3,' meters
     .')
 1460 FORMAT (/,18X,'Triangulated Object Coordinates will be saved')
 1470 FORMAT (/,16X,'Triangulated Object Coordinates will not be saved')
 1480 FORMAT (/,16X,'Adjusted Camera Station Parameters will be saved')
 1490 FORMAT (/,14X,'Adjusted Camera Station Parameters will not be save
     .d')
      END
```

```
      SUBROUTINE READIM (NFRM,LEADZ,ITAPE,JTAPE)
C
C  CONSTRUCT IMAGE DATA FILE AND ITS INDEX
C
      CHARACTER* 1LEADZ
      CHARACTER*80 INFM1,INFM2
      COMMON /TAPES/ IN,IO,IOS,IDUM(14)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'INDXFR.INC'
      INCLUDE 'RANVAR.INC'
      INCLUDE 'HPUNIX.INC'
      DIMENSION FOCALS(ISZ5), IDFOCL(2,ISZ5)
      DIMENSION IDS(4,100), XY(4,100)
      DIMENSION ID12(2), ID34(2)
      EQUIVALENCE (ID1,ID12(1)), (ID2,ID12(2)), (ID3,ID34(1)), (ID4,
     .ID34(2))
      EQUIVALENCE (IDS(1,1),XY(1,1),IBUF(1))
      DATA IEND/'****'/
      DATA NMAX,MMAX,MAXB/ISZ1,ISZ5,100/
      DATA ZERO/0.0/
      DATA INFM1/'(2A4,2X,F10.3)'/
      DATA INFM2/'(2A4,2X,2F10.3)'/
C
      OPEN (UNIT=10,ACCESS='DIRECT',FORM='UNFORMATTED',STATUS='SCRATCH',
     .RECL=1600)
C
C  Define input and output data sets
C
      INF1=ITAPE
      INF2=JTAPE
C
C  Read camera systems' principal distances
C
      CALL TSTFRM (INFM1,INF1,IND)
      IF (IND.EQ.0) GO TO 1030
      NCAM=0
 1010 READ (INF1,1230) ID1,ID2,F
      IF (ID1.EQ.IEND) GO TO 1030
      CALL REFRM (ID12,LEADZ)
      IF (NCAM.GT.MMAX) GO TO 1020
      NCAM=NCAM+1
      IDFOCL(1,NCAM)=ID1
      IDFOCL(2,NCAM)=ID2
      FOCALS(NCAM)=F
      GO TO 1010
 1020 CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1240) NCAM,MMAX
      STOP
C
C  Construct image data file
C
 1030 IP=2
```

```
      NFRM=0
      NB=1
      NP=0
      ITERM=0
      IEOF=0
      CALL TSTFRM (INFM2,INF2,IND)
      IF (IND.EQ.0) GO TO 1220
 1040 READ (INF2,1250,END=1170) ID1,ID2,F,SX,SY,ID3,ID4
      CALL REFRM (ID12,LEADZ)
      IF (F.NE.ZERO) GO TO 1080
      CALL REFRM (ID34,LEADZ)
      IF (NCAM.NE.0) GO TO 1050
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1260)
      STOP
 1050 DO 1060 II=1,NCAM
           IF (ID3.EQ.IDFOCL(1,II).AND.ID4.EQ.IDFOCL(2,II)) GO TO 1070
 1060 CONTINUE
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1270) ID3,ID4,ID1,ID2
      STOP
 1070 F=FOCALS(II)
 1080 IF (NFRM.EQ.0) GO TO 1100
      DO 1090 I=1,NFRM
           IF (ID1.NE.INDEXM(1,I).OR.ID2.NE.INDEXM(2,I)) GO TO 1090
           CALL CLR
           CALL TOPLFT
           CALL CURDWN (8)
           CALL BEEP
           WRITE (*,1280) ID1,ID2
           STOP
 1090 CONTINUE
      IF (NFRM.NE.NMAX) GO TO 1100
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1290) NFRM,NMAX
      STOP
 1100 NFRM=NFRM+1
      INDEXM(1,NFRM)=ID1
      INDEXM(2,NFRM)=ID2
      INDEXM(3,NFRM)=IP+32768*NB
      XY(1,NB)=F
      XY(2,NB)=SX
      XY(3,NB)=SY
      GO TO 1130
 1110 READ (INF2,INFM2,END=1180) ID3,ID4,X,Y
      CALL REFRM (ID34,LEADZ)
```

```
 1120 NP=NP+1
      IDS(1,NB)=ID3
      IDS(2,NB)=ID4
      XY(3,NB)=X
      XY(4,NB)=Y
 1130 NB=NB+1
      IF (NB.LE.MAXB) GO TO 1150
 1140 WRITE (10,REC=IP) IBUF
      IP=IP+1
      IF (ITERM.NE.0) GO TO 1190
      NB=1
 1150 IF (ID3.NE.IEND.AND.ID4.NE.IEND) GO TO 1110
      IF (IEOF.EQ.1) GO TO 1170
      IF (NP.GT.1) GO TO 1160
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1300) ID1,ID2
      STOP
 1160 NP=0
      GO TO 1040
 1170 IF (NB.EQ.1) GO TO 1190
      ITERM=1
      GO TO 1140
 1180 ID3=IEND
      IEOF=1
      GO TO 1120
 1190 IQ=IP
      K=1
      DO 1200 I=1,3
           DO 1200 J=1,NFRM
           IBUF(K)=INDEXM(I,J)
           K=K+1
           IF (K.LE.100) GO TO 1200
           WRITE (10,REC=IP) IBUF
           IP=IP+1
           K=1
 1200 CONTINUE
      IF (K.EQ.1) GO TO 1210
      WRITE (10,REC=IP) IBUF
      IP=IP+1
 1210 IP=1
      IBUF(1)=IQ
      IBUF(2)=NFRM
      WRITE (10,REC=IP) IBUF
 1220 RETURN
C
 1230 FORMAT (2A4,2X,F10.3)
 1240 FORMAT (20X,I3,' NUMBER OF CAMERA SYSTEMS EXCEEDED',I3)
 1250 FORMAT (2A4,2X,3F10.3,2A4)
 1260 FORMAT (//10X,'INPUT DOES NOT CONTAIN CAMERA FOCAL LENGTH(s)')
 1270 FORMAT (//10X,'UNRECOGNIZED CAMERA ID ',2A4,' FOR FRAME ',2A4)
 1280 FORMAT (//10X,'FRAME ',2A4,' IS INCLUDED IN INPUT MORE THAN ONCE')
 1290 FORMAT (//20X,I4,' CAMERA STATIONS EXCEEDED ',I4)
```

```
 1300 FORMAT (//20X,'NO IMAGE POINTS GIVEN FOR FRAME ',2A4)
      END

      SUBROUTINE TSTFRM (IFRM,IFILE,IND)
C
C  TEST RECORD IMAGES FOR FORMAT SPECIFICATIONS
C
      CHARACTER*1 IBLANK,IENDL,IENDR,IFRST,ILAST,ID
      CHARACTER*80 IDS,IFRM
      DATA IBLANK/' '/
      DATA IENDL/'('/
      DATA IENDR/')'/
C
C  Read candidate format and check its validity
C
      IND=0
C
      READ (IFILE,'(A80)',END=1050) IDS
      IND=1
      IFRST=IBLANK
      ILAST=IBLANK
      DO 1020 I=1,80
           ID=IDS(I:I)
           IF (ID.EQ.IBLANK) GO TO 1020
           ILAST=ID
           IF (IFRST.EQ.IBLANK) IFRST=ID
 1020 CONTINUE
      IF (IFRST.NE.IENDL.OR.ILAST.NE.IENDR) GO TO 1040
           IFRM=IDS
      GO TO 1050
 1040 BACKSPACE IFILE
 1050 RETURN
      END

      SUBROUTINE REFRM (IID,LEADZ)
C
C  COUNT LEADING BLANKS AND SPECIAL CHARACTERS
C
C  Note that LEADZ is input from the Options card.
C
      CHARACTER*8 ID,NEWID
      CHARACTER*1 LEADZ,BLANK,CH
      DIMENSION IID(2)
      DATA BLANK/' '/
C
C  Do the same as: ENCODE (8,1000,ID) IID
C
      WRITE (ID,1070) IID
      J=0
      NEWID=ID
      DO 1010 I=1,8
           CH=NEWID(I:I)
           IF (CH.NE.BLANK.AND.CH.NE.LEADZ) GO TO 1020
```

```
         J=I
 1010 CONTINUE
C
C  Count trailing blanks
C
 1020 K=0
      DO 1030 I=8,1,-1
         CH=NEWID(I:I)
         IF (CH.NE.BLANK) GO TO 1040
         K=9-I
 1030 CONTINUE
C
C  Right justify
C
 1040 I=8-J-K
      IF (I.GE.8) RETURN
      DO 1050 IP=1,8
         ID(IP:IP)=BLANK
 1050 CONTINUE
      IF (I.LE.0) THEN
         ID(8:8)=LEADZ
         GO TO 1060
      END IF
      J=J+1
      L=9-I
      ID(L:L+I-1)=NEWID(J:J+I-1)
C
C  Do the same as: DECODE (8,1000,ID) IID
C
 1060 READ (ID,1070) IID
      RETURN
C
C
 1070 FORMAT (2A4)
      END


      SUBROUTINE GETFR (ID,F,VARX,VARY)
C
C  RETRIEVE FRAME MEASUREMENTS
C
      INCLUDE 'PARAMS.INC'
      INCLUDE 'INDXFR.INC'
      INCLUDE 'RANVAR.INC'
      INCLUDE 'HPUNIX.INC'
      DIMENSION ID(2)
      DIMENSION IDS(4,100), XY(4,100)
      EQUIVALENCE (IDS(1,1),XY(1,1),IBUF(1))
      DATA INDX/1/
C
C  Test for first entry and load index array
C
      IF (INDX.EQ.0) GO TO 1030
      INDX=0
      IP=1
```

```
      READ (10,REC=IP) IBUF
      IP=IBUF(1)
      IQ=IP
      NFRM=IBUF(2)
      K=400
      DO 1020 I=1,3
           DO 1020 J=1,NFRM
           IF (K.LT.400) GO TO 1010
           K=0
           IP=IQ
           READ (10,REC=IP) IBUF
           IQ=IQ+1
 1010      K=K+1
           INDEXM(I,J)=IBUF(K)
 1020 CONTINUE
C
C  Extract principal distance and image variances
C
 1030 DO 1040 I=1,NFRM
           IF (ID(1).EQ.INDEXM(1,I).AND.ID(2).EQ.INDEXM(2,I)) GO TO
            1050
 1040 CONTINUE
C
C  Write error message:
C
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1070) ID
      STOP
 1050 J=INDEXM(3,I)
      NB=J/32768
      IP=J-32768*NB
      IQ=IP
      READ (10,REC=IP) IBUF
      IQ=IQ+1
      F=XY(1,NB)
      VARX=XY(2,NB)
      VARY=XY(3,NB)
      NB=NB+1
      RETURN
C
C  This entry extracts coordinates of one image point
C
      ENTRY GETPT(ID,X,Y)
C
      IF (NB.LE.100) GO TO 1060
      IP=IQ
      READ (10,REC=IP) IBUF
      IQ=IQ+1
      NB=1
 1060 ID(1)=IDS(1,NB)
      ID(2)=IDS(2,NB)
      X=XY(3,NB)
```

```
      Y=XY(4,NB)
      NB=NB+1
C
      RETURN
C
C
 1070 FORMAT (//20X,'COULD NOT LOCATE FRAME ',2A4,' IN IMAGE DATA FILE')
      END


      DOUBLE PRECISION FUNCTION DEGRAD (ANG)
C
C  TRANSFORM DMS ANGLE TO RADIANS
C
      IMPLICIT DOUBLEPRECISION(A-H,M-Z)
      DIMENSION CODE(2)
      DATA CODE/10000.0D0,100.0D0/
      DATA ZERO,ONE/0.0D0,1.0D0/
      DATA C1,C2/3600.0D0,60.0D0/
      PI=4.D0*(DATAN(1.D0))
      SECRAD=PI/180.D0/C1
C
C  Separate degree field
C
      FACTOR=ONE
      IF (ANG.LT.ZERO) FACTOR=-ONE
      SEC=DABS(ANG)
      TMP=CODE(1)
      I=SEC/TMP
      IF (I.GT.360) GO TO 1010
      DEG=I
C
C  Separate minutes field
C
      SEC=SEC-DEG*TMP
      TMP=CODE(2)
      I=SEC/TMP
      IF (I.GT.60) GO TO 1010
      MIN=I
C
C  Separate seconds field
C
      SEC=SEC-MIN*TMP
      IF (SEC.GT.C2) GO TO 1010
      SEC=SECRAD*(DEG*C1+MIN*C2+SEC)*FACTOR
      DEGRAD=SEC
      RETURN
C
C  Error detected in dms form
C
 1010 CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1020)
```

```
      STOP
C
C
 1020 FORMAT (' **** ILLEGAL DMS FIELD DETECTED IN INPUT STREAM ****')
      END


      SUBROUTINE RADDEG (RAD,DMS)
C
C  CONVERT ANGLE FROM RADIANS TO DMS
C
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
      CHARACTER*15 DMS
      CHARACTER*1 SIGN
      INTEGER ISEC,IDEG,IMIN
      DATA ZERO/0.0D0/
      PI=4.D0*(DATAN(1.D0))
      RADSEC=180.D0*3600.D0/PI
C
C  Determine the sign of angle
C
      SIGN=' '
      IF (RAD.EQ.ZERO) THEN
           IDEG=0
           IMIN=0
           SEC=0.
           GO TO 1010
      END IF
      IF (RAD.LT.ZERO) SIGN='-'
C
C Convert angle to seconds of arc
C
      SEC=DABS(RAD)*RADSEC
      ISEC=SEC
C
C Compute degrees, minutes,and seconds parts of angle
C
      IDEG=ISEC/3600
      ISEC=MOD(ISEC,3600)
      IMIN=ISEC/60
      SEC=SEC-IDEG*3600-IMIN*60
      IF (SEC.GE.59.99999) IMIN=IMIN+1
      IF (SEC.GE.59.99999) SEC=0.0D0
      IF (IMIN.EQ.60) IDEG=IDEG+1
      IF (IMIN.EQ.60) IMIN=0
C
C  Form dms character field
C
C  Write the equivalent of: ENCODE (15,1000,DMS) SIGN,IDEG,IMIN,SEC
C
 1010 WRITE (DMS,1020) SIGN,IDEG,IMIN,SEC
      RETURN
C1020 FORMAT (A1,2I3,F8.4)
C
 1020 FORMAT (A1,2I3.2,F8.4)
```

```
      END

      SUBROUTINE BLOCKD (ITAPE,JTAPE,KTAPE)
C
C  Read all images, sort them in ascending ident order,
C  and block them into records. Size of each record is dependent
C  upon the number of equal idents.
C
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
      INTEGER XY(2)
      DIMENSION PTSM(100), PTST(100), IMAGES(4,100), IDCAMM(100),
     .IDCAMT(100), ITABL(4,2100)
      EQUIVALENCE (CON,XY(1))
      DATA ITBMAX/2000/
C
C  ** ITAPE ** Scratch file
C  ** JTAPE ** Output blocked data file
C  ** KTAPE ** Input data from RDFRAM Subroutine
C
      IPASS=0
      REWIND KTAPE
 1010 REWIND ITAPE
      REWIND JTAPE
      MTBL=0
      IPASS=IPASS+1
C
C  Read images record and check for sentinel
C
 1020 READ (KTAPE) NIMG,((IMAGES(I,J),I=1,4),J=1,NIMG)
      IF (IMAGES(1,1).EQ.0) GO TO 1040
C
C  Insert the images into table
C
      DO 1030 I=1,NIMG
           MTBL=MTBL+1
           DO 1030 J=1,4
 1030      ITABL(J,MTBL)=IMAGES(J,I)
C
C  Check if the images table is full
C
      IF (MTBL.LE.ITBMAX) GO TO 1020
C
C  Check for any entries in images table
C
 1040 IF (MTBL.EQ.0) GO TO 1190
C
C  Sort the images in ascending ident order
C
      CALL SORT (ITABL,4,MTBL)
C
C  Check for first data pass. If not, begin to merge the
C  images with the previous blocked images.
C
      IF (IPASS.EQ.1) GO TO 1060
```

```
      ISWCH=1
 1050 READ (JTAPE) IDT,NPT,(IDCAMT(I),I=1,NPT),(PTST(I),I=1,NPT)
      GO TO (1060,1110,1130),ISWCH
C
C  Collect a block of images from table
C
 1060 NPH=1
 1070 NPL=NPH
 1080 IF (NPH.EQ.MTBL) GO TO 1090
      IF (ITABL(1,NPH).NE.ITABL(1,NPH+1)) GO TO 1090
      NPH=NPH+1
      GO TO 1080
 1090 NPM=NPH+1-NPL
      IDM=ITABL(1,NPH)
      DO 1100 I=1,NPM
           XY(1)=ITABL(2,NPL)
           XY(2)=ITABL(3,NPL)
           PTSM(I)=CON
           IDCAMM(I)=ITABL(4,NPL)
 1100      NPL=NPL+1
C
C  A table block has been collected. Check for first data pass.
C
      IF (IPASS.EQ.1) GO TO 1120
C
C  Not first data pass; check for tape blocks exhaustion.
C
 1110 IF (IDT.EQ.0) GO TO 1160
C
C  Tape blocks not exhausted; check for table exhaustion.
C
      IF (NPH.GT.MTBL) GO TO 1150
C
C  Test the ident of the table block against the ident
C  of the tape block.
C
      IF (IDM-IDT) 1120,1140,1150
C
C  Ident of table block is less. write the table block
C  onto tape and check if table is exhausted.
C
 1120 WRITE (ITAPE) IDM,NPM,(IDCAMM(I),I=1,NPM),(PTSM(I),I=1,NPM)
      NPH=NPH+1
 1130 IF (NPH.GT.MTBL) GO TO 1170
      GO TO 1070
C
C  The idents of the table block and the tape block are equal.
C  merge and write them onto tape.
C
 1140 ISUM=NPM+NPT
      WRITE (ITAPE) IDM,ISUM,(IDCAMM(I),I=1,NPM),(IDCAMT(I),I=1,NPT),
     .(PTSM(I),I=1,NPM),(PTST(I),I=1,NPT)
      NPH=NPH+1
      ISWCH=3
      GO TO 1050
```

```
C
C  Ident of table block is greater. write the tape block onto tape.
C
 1150 WRITE (ITAPE) IDT,NPT,(IDCAMT(I),I=1,NPT),(PTST(I),I=1,NPT)
      ISWCH=2
      GO TO 1050
C
C  Tape blocks is exhausted. Check for table exhaustion.
C
 1160 IF (NPH.GT.MTBL) GO TO 1180
      GO TO 1120
C
C  Table is exhausted. Check if first data pass.
C  If not, check for tape blocks exhaustion.
C
 1170 IF (IPASS.EQ.1) GO TO 1180
      IF (IDT.NE.0) GO TO 1150
C
C  Write a sentinel onto output tape.
C
 1180 IDM=0
      NPM=1
      WRITE (ITAPE) IDM,NPM,IDCAMM(1),PTSM(1)
C
C  Alternate tapes for next data pass - if necessary.
C
      I=JTAPE
      JTAPE=ITAPE
      ITAPE=I
C
C  Check for the presence of more images. If present, repeat
C  the process for the next data pass.
C
      IF (IMAGES(1,1).NE.0) GO TO 1010
C
 1190 RETURN
      END


      SUBROUTINE SORT (ITABL,NR,NC)
C
C  SORT A TWO DIMENSIONAL ARRAY ITABL(NR,NC) ON THE DATA of row 1.
C
      DIMENSION ITABL(NR,NC)
C
      IF (NC.LE.1) RETURN
      NCM=NC-1
      DO 1030 I=1,NCM
           MINM1=ITABL(1,I)
           IN=I
           IP=I+1
           DO 1010 J=IP,NC
                IVAL1=ITABL(1,J)
                IF (IVAL1.GE.MINM1) GO TO 1010
                MINM1=IVAL1
```

```
              IN=J
 1010     CONTINUE
          IF (IN.EQ.I) GO TO 1030
          DO 1020 KK=1,NR
              ITEMP=ITABL(KK,I)
              ITABL(KK,I)=ITABL(KK,IN)
 1020         ITABL(KK,IN)=ITEMP
 1030 CONTINUE
      RETURN
      END

      SUBROUTINE MERGEG (ITAPE,JTAPE,KTAPE,LTAPE,MTAPE)
C
C THIS PROGRAM MERGES THE OBJECT CONTROL WITH THE BLOCKED
C IMAGES AND FORMS THE DATA TAPE FOR THE CAMERA STATIONS
C TRIANGULATION PROCESS.
C
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
      REAL*4 PTSP(2,100)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK11.INC'
      INCLUDE 'GPCTRS.INC'
      INCLUDE 'OPTION.INC'
      INCLUDE 'OPTON2.INC'
C
      DIMENSION ICAMTB(ISZ1)
      DIMENSION IDCAMB(100), ICNTRL(300), IPASPT(500), MAXTEN(100)
      DIMENSION PTS(100), CONTRL(9)
      DIMENSION ZEROM(6)
C
      DIMENSION GCPTS(6,ISZ3), INDXP(3,ISZ3)
      DIMENSION IMAGES(4,100)
      EQUIVALENCE (PTS(1),PTSP(1,1)), (IMAGES(1,1),INDXP(1,1))
C
      DATA MAXBLK,MS1,MS2,MS3,NCCTR,NGCTR/ISZ4,300,500,100,1,1/
      DATA MAXCTR,ICNCTR,IPSCTR/0,0,0/
      DATA ZEROM/6*0.0D0/
      DATA IPONE,IMONE/1,-1/
C
C PASS OVER THE IMAGES.
C
C ** ITAPE ** OUTPUT POINTER FILE
C ** JTAPE ** OUTPUT BLOCKED OBJECT DATA FILE
C ** KTAPE ** INPUT / OUTPUT CAMERA PARAMETERS
C ** LTAPE ** INPUT BLOCKED DATA FROM BLOCKD SUBROUTINE
C ** MTAPE ** OUTPUT OBJECT IDENTIFICATIONS
C
      REWIND ITAPE
      REWIND JTAPE
      REWIND LTAPE
      REWIND MTAPE
      NGPS=0
      NIND=0
C
```

```
C READ CAMERA STATIONS DATA.
C
      READ (KTAPE) N,((PARAM(I,J),I=1,6),J=1,N),((VARPLT(I,J),I=1,2),J=
     .1,N),(FOCAL(I),I=1,N),((WTMAT(I,J),I=1,6),J=1,N),((IDCAM(I,J),I=1,
     .2),J=1,N)
      READ (KTAPE) M,((IDPLT(I,J),I=1,2),J=1,M)
      READ (KTAPE) N,((INDEX(I,J),I=1,2),J=1,N)
C
C READ OBJECT CONTROL DATA.
C
      READ (KTAPE) NG,((INDXP(I,J),I=1,3),J=1,NG),((GCPTS(I,J),I=1,6),J=
     .1,NG)
      REWIND KTAPE
C
C SORT CAMERA AND OBJECT CONTROL INDICES.
C
      CALL SORT (INDEX,2,N)
      IF (NG.NE.0) CALL SORT (INDXP,3,NG)
C
C CLEAR INTEGER CAMERA IDENTIFICATION TABLE.
C
      DO 1010 I=1,N
 1010     ICAMTB(I)=0
C
C READ BLOCKED IMAGES RECORD. CHECK FOR DATA SENTINEL.
C
 1020 READ (LTAPE) IDBLK,NIMG,(IDCAMB(I),I=1,NIMG),(PTS(I),I=1,NIMG)
      IF (IDBLK.NE.0) GO TO 1030
      IDBLK=1073741825
      GO TO 1120
C
C ELIMINATE DUPLICATE PLATE MEASUREMENTS.
C
 1030 NN=0
      DO 1050 I=1,NIMG
          ID=IDCAMB(I)
          IF (ID.EQ.0) GO TO 1050
          NN=NN+1
          IDCAMB(NN)=ID
          PTS(NN)=PTS(I)
          IF (I.EQ.NIMG) GO TO 1050
          MM=1
          LL=I+1
          DO 1040 J=LL,NIMG
              IF (ID.NE.IDCAMB(J)) GO TO 1040
              MM=MM+1
              IDCAMB(J)=0
              PTSP(1,NN)=PTSP(1,NN)+PTSP(1,J)
              PTSP(2,NN)=PTSP(2,NN)+PTSP(2,J)
 1040     CONTINUE
          IF (MM.EQ.1) GO TO 1050
          PTSP(1,NN)=PTSP(1,NN)/FLOAT(MM)
          PTSP(2,NN)=PTSP(2,NN)/FLOAT(MM)
 1050 CONTINUE
      NIMG=NN
```

```
C
C CHECK ON MAXIMUM SIZE OF BLOCK.
C
      IF (NIMG.LE.MAXBLK) GO TO 1060
      NIMG=MAXBLK
      IF (MAXCTR.EQ.MS3) GO TO 1060
      MAXCTR=MAXCTR+1
      MAXTEN(MAXCTR)=IDBLK
C
C DETERMINE IF BLOCK HAS CORRESPONDING CONTROL POINT.
C
 1060 IND=7
      IF (NGCTR.GT.NG.OR.IDBLK.LT.INDXP(1,NGCTR)) GO TO 1080
      IND=INDXP(3,NGCTR)
      I=INDXP(2,NGCTR)
      DO 1070 J=1,6
 1070     CONTRL(J)=GCPTS(J,I)
      NGCTR=NGCTR+1
C
C CHECK ON MINIMUM SIZE OF BLOCK.
C
 1080 IF (NIMG.GT.1) GO TO 1100
      IF (IND.EQ.7) GO TO 1090
      IF (IND.LT.3.OR.IND.EQ.4) GO TO 1100
      IF (ICNCTR.EQ.MS1) GO TO 1020
      ICNCTR=ICNCTR+1
      ICNTRL(ICNCTR)=IDBLK
      GO TO 1020
 1090 IF (IPSCTR.EQ.MS2) GO TO 1020
      IPSCTR=IPSCTR+1
      IPASPT(IPSCTR)=IDBLK
      GO TO 1020
C
C CHECK TO WRITE A RECORD FOR THE FIRST APPEARANCE
C OF EACH INTEGER CAMERA IDENTIFICATION.
C
 1100 DO 1110 I=1,NIMG
          J=IDCAMB(I)
          IF (ICAMTB(J).NE.0) GO TO 1110
          ICAMTB(J)=IPONE
          J=-J
          WRITE (ITAPE) IDBLK,J,IPONE
          NIND=NIND+1
 1110 CONTINUE
C
C WRITE MERGED BLOCKED IMAGES/OBJECT CONTROL.
C
      WRITE (ITAPE) IDBLK,NIMG,IND
      WRITE (JTAPE) (IDCAMB(I),I=1,NIMG),(CONTRL(I),I=1,6),ZEROM,(PTS(I)
     .,I=1,NIMG)
      NIND=NIND+1
      NGPS=NGPS+1
C
C CHECK TO WRITE A DELETION RECORD.
C
```

```
 1120 IF (NCCTR.GT.N) GO TO 1130
      IF (IDBLK.LT.INDEX(1,NCCTR)) GO TO 1020
      I=-INDEX(2,NCCTR)
      WRITE (ITAPE) IDBLK,I,IMONE
      NIND=NIND+1
      NCCTR=NCCTR+1
      IF (IDBLK.EQ.1073741825) GO TO 1120
      GO TO 1020
C
C PROCESSING OF THE BLOCKS IS FINISHED.
C WRITE A SENTINEL RECORD.
C
 1130 I=0
      REWIND LTAPE
      WRITE (ITAPE) IDBLK,I,IND
C
C WRITE OBJECT POINT IDENTS.
C
      WRITE (MTAPE) M,((IDPLT(I,J),I=1,2),J=1,M)
      REWIND MTAPE
C
C CHECK TO LIST CONTROL POINTS APPEARING
C ON ONE PHOTOGRAPH ONLY.
C
      IF (ICNCTR.NE.0) CALL PRINTM (ICNTRL,ICNCTR,1)
C
C CHECK TO LIST PASS-POINTS APPEARING
C ON ONE PHOTOGRAPH ONLY.
C
      IF (IPSCTR.NE.0) CALL PRINTM (IPASPT,IPSCTR,2)
C
C CHECK TO LIST PASS-POINTS APPEARING
C ON MORE THAN TEN PHOTOGRAPHS.
C
      IF (MAXCTR.NE.0) CALL PRINTM (MAXTEN,MAXCTR,3)
C
C STORE CAMERA PARAMETERS.
C
      REWIND KTAPE
      WRITE (KTAPE) N,((PARAM(I,J),I=1,6),J=1,N),((VARPLT(I,J),I=1,2),J=
     .1,N),(FOCAL(I),I=1,N),((WTMAT(I,J),I=1,6),J=1,N),((IDCAM(I,J),I=1,
     .2),J=1,N)
      REWIND KTAPE
C
      RETURN
      END


      SUBROUTINE PRINTM (IDS,ICTR,ISWCH)
C
C THIS PROGRAM LISTS THE WARNING MESSAGES FOR PHASE1.
C
      REAL*8 PARAM
      COMMON /TAPES/ IN,IO,IOS,IDUM(14)
      INCLUDE 'PARAMS.INC'
```

```
      INCLUDE 'WORK11.INC'
      INCLUDE 'WARNGS.INC'
      DIMENSION IDS(50), IMAGES(2,4)
C
C CHECK TO LIST THE PAGE HEADING.
C
      IF (IERR.NE.0) GO TO 1010
      CALL NEWPAG
      WRITE (IO,1070)
      WRITE (IOS,1120)
C
C LIST ERROR WARNINGS TITLE.
C
 1010 GO TO (1020,1030,1040),ISWCH
 1020 WRITE (IO,1080)
      WRITE (IOS,1130)
      GO TO 1050
 1030 WRITE (IO,1090)
      WRITE (IOS,1140)
      GO TO 1050
 1040 WRITE (IO,1100)
      WRITE (IOS,1150)
C
C LIST THE IDENTS OF THE POINTS.
C
 1050 J=0
   \  DO 1060 I=1,ICTR
           J=J+1
           ID=IDS(I)
           IMAGES(1,J)=IDPLT(1,ID)
           IMAGES(2,J)=IDPLT(2,ID)
           IF (J.NE.4) GO TO 1060
           WRITE (IO,1110) (IMAGES(1,J),IMAGES(2,J),J=1,4)
           WRITE (IOS,1160) (IMAGES(1,J),IMAGES(2,J),J=1,4)
           J=0
 1060 CONTINUE
      IF (J.NE.0) WRITE (IO,1110) (IMAGES(1,I),IMAGES(2,I),I=1,J)
      IF (J.NE.0) WRITE (IOS,1160) (IMAGES(1,I),IMAGES(2,I),I=1,J)
      IERR=1
C
      RETURN
C
C
 1070 FORMAT (51X,'E R R O R   W A R N I N G S'/)
 1080 FORMAT (///48X,'CONTROL POINTS APPEARING ON 1 PHOTO'/)
 1090 FORMAT (///50X,'PASS POINTS APPEARING ON 1 PHOTO'/)
 1100 FORMAT (///44X,'PASS POINTS APPEARING ON MORE THAN 10 PHOTOS'/)
 1110 FORMAT (40X,4(4X,2A4))
 1120 FORMAT (30X,'E R R O R   W A R N I N G S'/)
 1130 FORMAT (///27X,'CONTROL POINTS APPEARING ON 1 PHOTO'/)
 1140 FORMAT (///29X,'PASS POINTS APPEARING ON 1 PHOTO'/)
 1150 FORMAT (///23X,'PASS POINTS APPEARING ON MORE THAN 10 PHOTOS'/)
 1160 FORMAT (19X,4(4X,2A4))
      END
```

```
      SUBROUTINE BEEP
C
C  THIS ROUTINE CAUSES A "BEEP" SOUND WHEN CALLED.
C
C  NOTE THAT THIS ROUTINE REQUIRES AN "ANSI TERMINAL"
C
      CHARACTER*1 BEEEP
      INTRINSIC CHAR
      BEEEP=CHAR(7)
      WRITE (*,'(1X,A1)') BEEEP
      RETURN
      END


      SUBROUTINE CLR
C
C  THIS ROUTINE ERASES ALL OF THE SCREEN AND THE CURSOR GOES TO
C  THE HOME POSITION.
C
C  NOTE THAT THIS ROUTINE REQUIRES AN "ANSI TERMINAL".
C
C     STRING = ESC [ 2 J
C
      CHARACTER*1 ESC,BKT,TWO,J
      CHARACTER*4 STRING
  \   ESC=CHAR(27)
      BKT=CHAR(91)
      TWO=CHAR(50)
      J=CHAR(74)
      STRING=ESC//BKT//TWO//J
      WRITE (*,'(1X,A4)') STRING
      RETURN
      END


      SUBROUTINE CURDWN (IROW)
C
C  THIS ROUTINE MOVES THE CURSOR DOWN ONE LINE WITHOUT CHANGING
C  COLUMNS.  THE VALUE OF IROW DETERMINES THE NUMBER OF LINES
C  MOVED.  THIS COMMAND IS IGNORED IF THE CURSOR IS ALREADY AT
C  THE BOTTOM OF THE SCREEN.
C
C    NOTE THAT THIS ROUTINE REQUIRES AN "ANSI TERMINAL"
C
      CHARACTER*1 ESC,BKT,B
      CHARACTER*2 ESCBKT
      ESC=CHAR(27)
      BKT=CHAR(91)
      ESCBKT=ESC//BKT
      B=CHAR(66)
      IF (IROW.LT.10) WRITE (*,'(1X,A2,I1,A1,/)') ESCBKT,IROW,B
      IF (IROW.GE.10) WRITE (*,'(1X,A2,I2,A1,/)') ESCBKT,IROW,B
      RETURN
      END
```

```
      SUBROUTINE TOPLFT
C
C  THIS SUBROUTINE MOVES THE CURSOR TO THE TOP LEFT OF THE SCROLLING
C  REGION.  THE ASSUMPTION IS THAT AN "ANSI" TERMINAL IS BEING USED.
C
      CHARACTER*1 ESCAPE,L_BRACKET,SEMICOLON,H
      CHARACTER*2 ESCBKT
      ESCAPE=CHAR(27)
      L_BRACKET=CHAR(91)
      ESCBKT=ESCAPE//L_BRACKET
      SEMICOLON=CHAR(59)
      H=CHAR(72)
      N=1
      WRITE (*,'(1X,A2,I1,A1,I1,A1,/)') ESCBKT,N,SEMICOLON,N,H
      RETURN
      END
```

# PC Giant

## Source Code

## File Name: 2.FOR (Calculations)

## 14 June 1990

```
      SUBROUTINE PHASE2
C
C THIS IS THE MAIN CALLING ROUTINE FOR LEAST SQUARES ADJUSTMENT
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*15    IDMSS,IDMS1,IDMS2
      CHARACTER*19 IOFM1
      DATA IOFM1/'(2A4,3F12.3,3G10.4)'/
      INCLUDE 'PARAMS.INC'
      INCLUDE 'TAPES.INC'
      INCLUDE 'WORK21.INC'
      INCLUDE 'WORK22.INC'
      INCLUDE 'WORK24.INC'
      INCLUDE 'UNITVR.INC'
      INCLUDE 'OPTION.INC'
      INCLUDE 'OPTON2.INC'
      INCLUDE 'CONVCR.INC'
C
      DIMENSION         TP(6), TW(6)
      DIMENSION         IDMSS(1,3), IDMS1(1), IDMS2(1)
C
      EQUIVALENCE       (IDMSS(1,1),IDMS1(1)), (IDMSS(1,2),IDMS2(1))
C
      DATA IE1,IE2      /ISZ8,ISZ9/
      DATA ZERO,ONE     /0.0D0,1.0D0/
C
C  LOAD INPUT CAMERA PARAMETERS
C
      REWIND ITAPE3
      READ (ITAPE3) NCAM,((PARAM (I,J),I=1,6),J=1,NCAM),
     .                   ((VARPLT(I,J),I=1,2),J=1,NCAM),
     .                              (FOCAL(I),I=1,NCAM),
     .                   ((WTMAT (I,J),I=1,6),J=1,NCAM),
     .                   ((IFOTO(I,J),I=1,2),J=1,NCAM)
      REWIND ITAPE3
      I=6*NCAM
      J=3*NCAM
      CALL FILL (SOLUTM,I,ZERO)
      CALL FILL (ACCSOL,J,ZERO)
C
C  ESTIMATE MISSING COORDINATES FOR OBJECT POINTS
C
      NMAX=ISZ1
      CALL INITID
      CALL MISCOM (ITAPE1,ITAPE2,ITAPE3)
      IF (ITRNG.NE.0) GO TO 1090
      NMAX=ISZ6
C
C  PERFORM LEAST SQUARES ADJUSTMENT OF THE TRIANGULATION NETWORK
C
      CALL NEWPAG
      WRITE (IO,1110)
      WRITE (IOS,2110)
      IF(IUNIT.EQ.0) WRITE (IOS,2111)
      IF(IUNIT.EQ.1) WRITE (IOS,2112)
```

```
      SSP=1.0D30
      DO 1020 II=1,NIT
C
C  INITIALIZE NORMAL EQUATIONS
C
           CALL INITID
           CALL FILL (EQN,IE1,ZERO)
           CALL FILL (CONV,IE2,ZERO)
C
C  PERFORM FORWARD SOLUTION
C
           CALL LEASTQ (ITAPE1,ITAPE2,ITAPE4,ITAPE5)
C
C  PERFORM BACKWARD SOLUTION
C
           CALL BACKSL (ITAPE5,ITAPE7)
           CALL UPDATG (ITAPE1,ITAPE2,ITAPE3,ITAPE4)
C
C  PRINT CAMERA CORRECTIONS
C
           WRITE (IO,1120) II
           WRITE (IOS,1130) II
           WRITE (*,1135) II
           DO 1010 I=1,NCAM
                ID1=IFOTO(1,I)
                ID2=IFOTO(2,I)
                IF (IUNIT.EQ.0) THEN
                      WRITE (IO,1140) ID1,ID2,(SOLUTM(J,I),J=1,6)
                      WRITE (IOS,2140) ID1,ID2,(SOLUTM(J,I),J=1,6)
                      ELSE
                      WRITE (IO,1150) ID1,ID2,(SOLUTM(J,I),J=1,6)
                      WRITE (IOS,2150) ID1,ID2,(SOLUTM(J,I),J=1,6)
                ENDIF
 1010      CONTINUE
C
C  WRITE "SS" Sum of the Squares to Screen & UNIT=IO:
C
           WRITE (IO,1160) SS
           WRITE (*,1170) SS
           WRITE (IOS,1170) SS
C
C TEST FOR CONVERGENCE
C
           CON=ONE-SS/SSP
           IF (DABS(CON).LE.EPSLN.OR.SS.LE.DFLOAT(IDFREE)) GO TO 1040
           IF (SS .GT. 1.1 * SSP) GO TO 1030
           SSP=SS
 1020 CONTINUE
C
C    CONVERGENCE FAILURE; WRITE BAD NEWS TO SCREEN & TO UNIT=IO:
C
 1030 CONTINUE
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
```

```
      CALL BEEP
      WRITE (IO,1180)
      WRITE (*,1190)
      WRITE (IOS,1190)
      IPROP=0
C
C  PUNCH CAMERA PARAMETERS
C
 1040 DO 1080 J=1,NCAM
              ID1=IFOTO(1,J)
              ID2=IFOTO(2,J)
              DO 1050 I=1,6
                     TP(I)=PARAM(I,J)
                     TW(I)=SQRT(ONE/WTMAT(I,J))
 1050         CONTINUE
              IF (IUNIT.EQ.0) GO TO 1060
              CALL RADDEG (TP(1),IDMS1)
              CALL RADDEG (TP(2),IDMS2)
              TP(1)=PAKDMS(IDMS1)
              TP(2)=PAKDMS(IDMS2)
              CALL RADDEG (TW(1),IDMS1)
              CALL RADDEG (TW(2),IDMS2)
              TW(1)=PAKDMS(IDMS1)
              TW(2)=PAKDMS(IDMS2)
 1060         WRITE (ITAPE0,IOFM1) ID1,ID2,(TP(K),K=1,3),(TW(K),K=1,3)
              DO 1070 K=1,3
                     L=K+3
                     CALL RADDEG (TP(L),IDMSS(1,K))
                     TP(K)=PAKDMS(IDMSS(1,K))
                     CALL RADDEG (TW(L),IDMSS(1,K))
                     TW(K)=PAKDMS(IDMSS(1,K))
 1070         CONTINUE
              WRITE (ITAPE0,IOFM1) ID1,ID2,(TP(K),K=1,3),(TW(K),K=1,3)
 1080 CONTINUE
 1090 CALL LSTPLR (ITAPE1,ITAPE2,ITAPE6,ITAPE3)
      IF (IPROP.EQ.0) GO TO 1100
      CALL PERROR (ITAPE1,ITAPE4,ITAPE7,ITAPE3,ITAPE2,ITAPE5)
      I=ITAPE2
      ITAPE2=ITAPE3
      ITAPE3=I
C
C  SAVE CAMERA PARAMETERS
C
 1100 REWIND ITAPE2
      WRITE (ITAPE2) NCAM,((PARAM(I,J),I=1,6),J=1,NCAM),
     .                    ((IFOTO(I,J),I=1,2),J=1,NCAM)
      REWIND ITAPE2
C
      RETURN
C
 1110 FORMAT (39X,'C A M E R A   S T A T I O N S   C O R R E C T I O N
     . S')
 1120 FORMAT (/61X,'ITERATION ',I3)
 1140 FORMAT (10X,2A4,'   POSITION ',3F9.4,' m.   ATTITUDE ',3F14.9)
 1150 FORMAT (10X,2A4,'   POSITION ',2F13.9,F10.1,'   ATTITUDE ',3F14.9)
```

```
 1160 FORMAT (/39X,'PROVISIONAL WEIGHTED SUM OF SQUARES = ',G13.6)
 1180 FORMAT (//,55X,'**** CONVERGENCE FAILURE ****')
 2110 FORMAT (13X,'C A M E R A   S T A T I O N S   C O R R E C T I O N
     . S'//11X,'------- P O S I T I O N -------',3X,
     .    '---------- A T T I T U D E ------'/)
 2111 FORMAT (15X,'X',9X,'Y',9X,'Z',14X,'Omega',6X,'Phi',6X,'Kappa')
 2112 FORMAT (12X,'Lng',7X,'Lat',7X,'Elv',13X,'Omega',6X,'Phi',6X,
     .          'Kappa')
 1130 FORMAT (/38X,'Iteration ',I3)
 1135 FORMAT (/34X,'Iteration ',I3)
 2140 FORMAT (1X,2A4,3F10.4,' m.',4X,3F10.6)
 2150 FORMAT (1X,2A4,2X,3(F9.3,2X),X,3(F10.7,2X))
 1170 FORMAT (/15X,'Provisional Weighted Sum of Squares = ',G13.6)
 1190 FORMAT (//30X,'**** CONVERGENCE FAILURE ****')
      END


      SUBROUTINE INITID
C
C  SUBROUTINE TO INITIALIZE INTERNAL CAMERA STATION IDENTIFICATIONS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK24.INC'
C
      DO 1010 I=1,NMAX
           IDCAM(I)=0
 1010 CONTINUE
      RETURN
C
C  ENTRY DROPID(ID) to eliminate camera station ID from internal list:
C
      ENTRY DROPID(ID)
C
              CALL LOCTID (ID,I)
              IDCAM(I)=0
C
      RETURN
      END


      SUBROUTINE LOCTID (ID,K)
C
C  EXTRACT THE CAMERA POSITION INTEGER (K) WHICH
C  CORRESPONDS TO THE CAMERA IDENTIFICATION (ID)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK24.INC'
C
      DO 1010 I=1,NMAX
           IDD=IDCAM(I)
           IF (IDD.NE.ID) GO TO 1010
           K=I
           RETURN
```

```
 1010 CONTINUE
C
C  WRITE ERROR MESSAGE ERROR IN LOCTID:
C
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1030) ID,I,IDCAM(I)
      STOP
C
 1030 FORMAT (' ','ERROR IN LOCTID: ID = ',I2,' IDCAM(',I2,') = ',I10)
      END


      SUBROUTINE DROP (ID,ITAPE)
C
C  ELIMINATE THE CAMERA STATION ID FROM THE NORMALS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK22.INC'
      INCLUDE 'WORK24.INC'
      COMMON /WORK23/ PIVT(6,6),SV1(6),SV2(6),TMP1(6,6),TMP2(72),
     .                ZEROM(36),XDUM(18,ISZ4),IDUM(3,ISZ4),IDUM2(3)
      DIMENSION       ISV1(6), ISV2(6)
C
      DO 1010 I=1,36
           ZEROM(I)=0.0D0
 1010 CONTINUE
      ONEM=-1.0D0
C
C  FORM TABLE OF CAMERA IDENTIFICATIONS
C
      N=0
      DO 1030 I=1,NMAX
           IDD=IDCAM(I)
           IF (IDD.EQ.0) GO TO 1030
           IF (IDD.NE.ID) GO TO 1020
           M=I
           GO TO 1030
 1020      N=N+1
           IDS(N)=IDD
 1030 CONTINUE
C
C  EXTRACT PIVOT MATRIX AND INVERT IT
C
      IDBLK=ID+32768*ID
      CALL STSUBM (PIVT,IDBLK,-1)
      CALL STSUBM (ZEROM,IDBLK,0)
      CALL INVRT (PIVT,6,ISV1,ISV2,6)
C
C  EXTRACT CONSTANT TERM
C
      CALL STSUBV (SV1,ID,-1)
```

```
      CALL STSUBV (ZEROM,ID,0)
C
C  EXTRACT CORRELATION MATRICES
C
      IF (N.EQ.0) GO TO 1050
      DO 1040 I=1,N
          IDBLK=IDS(I)+32768*ID
          CALL STSUBM (TMPST(1,I),IDBLK,-1)
          CALL STSUBM (ZEROM,IDBLK,0)
 1040 CONTINUE
C
C  ZERO CAMERA ID
C
 1050 IDCAM(M)=0
C
C  STORE THE DATA FOR BACK SUBSTITUTION
C
      M=N
      IF (M.EQ.0) M=1
      WRITE (ITAPE) N,M,ID,IDS,PIVT,SV1,((TMPST(I,J),I=1,36),J=1,M)
      IF (N.EQ.0) GO TO 1070
C
C  PERFORM ELIMINATION PROCESS
C
      CALL MPYAB (PIVT,ONEM,PIVT,36,1,1)
      DO 1060 I=1,N
          CALL MPYAB (TMPST(1,I),PIVT,TMP1,6,6,6)
          CALL MPYAB (TMP1,SV1,SV2,6,6,1)
          IDD=IDS(I)
          CALL STSUBV (SV2,IDD,1)
          DO 1060 J=I,N
          CALL MPYABT (TMP1,TMPST(1,J),TMP2,6,6,6)
          IDBLK=IDD+32768*IDS(J)
          CALL STSUBM (TMP2,IDBLK,1)
 1060 CONTINUE
C
 1070 RETURN
      END


      SUBROUTINE MISCOM (ITAPE,JTAPE,KTAPE)
C
C  ESTIMATE MISSING COMPONENTS OF OBJECT POINTS
C    and/or RESIDUALS OF PLATE COORDINATES.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK21.INC'
      REAL*4 PT
      COMMON /WORK23/ GXYZ(3),DXYZ(3,3),EQN(3,3),CV(3),AM(2,3),TMP1(3,3)
     .         ,TMP2(2,3),VEC(3),V(2),OBJECT(3,4),PT(2,ISZ4),IDCAM(ISZ4)
     .         ,XDUM(130),XDUM2(18,ISZ4),IDUM(3)
      INCLUDE 'WORK25.INC'
      INCLUDE 'SWITCH.INC'
      INCLUDE 'OPTION.INC'
```

```
      INCLUDE 'UNITVR.INC'
      INCLUDE 'OPTON2.INC'
      DIMENSION ITMP1(1), ITMP2(1)
      EQUIVALENCE (ITMP1(1),TMP1(1,1)), (ITMP2(1),TMP2(1,1))
      DATA ZERO/0.0D0/
C
C  INITIALIZATION
C  ** ITAPE ** POINTERS FILE
C  ** JTAPE ** INPUT BLOCKED OBJECT DATA FILE
C  ** KTAPE ** OUTPUT BLOCKED OBJECT DATA FILE
C
      IS=0
      IDFREE=0
      IF (IWGHT.EQ.0) IDFREE=-6*NCAM
      CALL INITID
      REWIND ITAPE
      REWIND JTAPE
      REWIND KTAPE
      AM(1,2)=ZERO
      AM(2,1)=ZERO
C
C  READ INDEX RECORD
C
 1010 READ (ITAPE) ID,NP,IND
      IF (NP) 1020,1110,1050
 1020 NP=-NP
      IF (IND.LT.0) GO TO 1040
      CALL MODID (NP)
      CALL LOCTID (NP,ID)
      CALL ROTMAT (PARAM(1,NP),R(1,1,ID),DXYZ,DXYZ,RL(1,1,ID))
      IF (IUNIT.EQ.0) GO TO 1030
      CALL PLHXYZ (PARAM(1,NP),STATON(1,ID),DXYZ)
      GO TO 1010
 1030 CALL COPY (PARAM(1,NP),STATON(1,ID),3)
      GO TO 1010
 1040 CALL DROPID (NP)
      GO TO 1010
 1050 READ (JTAPE) (IDCAM(I),I=1,NP),OBJECT,((PT(I,J),I=1,2),J=1,NP)
      IDFREE=IDFREE+2*NP
C
C  INITIALIZE NORMAL EQUATIONS
C
      CALL FILL (EQN,9,ZERO)
      CALL FILL (CV,3,ZERO)
C
C  FORM NORMAL EQUATIONS
C
      DO 1060 II=1,NP
           IDC=IDCAM(II)
           CALL LOCTID (IDC,ID)
           AM(1,1)=FOCAL(IDC)
           AM(1,3)=-PT(1,II)
           AM(2,2)=AM(1,1)
           AM(2,3)=-PT(2,II)
           CALL MPYABT (AM,R(1,1,ID),TMP2,2,3,3)
```

```
            CALL MPYAB  (TMP2,STATON(1,ID),V,2,3,1)
            CALL MPYATB (TMP2,TMP2,TMP1,3,2,3)
            CALL ADDMAT (EQN,TMP1,EQN,9)
            CALL MPYATB (TMP2,V,VEC,3,2,1)
            CALL ADDMAT (CV,VEC,CV,3)
 1060 CONTINUE
C
C   SOLVE FOR OBJECT COORDINATES
C
      CALL INVRT (EQN,3,ITMP1,ITMP2,3)
      CALL MPYAB (EQN,CV,VEC,3,3,1)
C
C   MODIFY MISSING COMPONENTS
C
      IF (IUNIT.EQ.0) GO TO 1070
      CALL XYZPLH (VEC,CV)
      GO TO 1080
 1070 CALL COPY (VEC,CV,3)
 1080 INDD=IND
      DO 1100 I=1,3
           ICODE=MOD(INDD,2)
           INDD=INDD/2
           IF (ICODE.EQ.0) GO TO 1090
           IDFREE=IDFREE-1
           OBJECT(I,1)=CV(I)
           OBJECT(I,2)=ZERO
           GO TO 1100
 1090      OBJECT(I,4)=OBJECT(I,1)-CV(I)
 1100 CONTINUE
C
C   WRITE MODIFIED OBJECT POINT RECORD
C
      WRITE (KTAPE) (IDCAM(I),I=1,NP),OBJECT,((PT(I,J),I=1,2),J=1,NP)
      GO TO 1010
C
 1110 I=JTAPE
      JTAPE=KTAPE
      KTAPE=I
      REWIND ITAPE
      REWIND JTAPE
      REWIND KTAPE
C
      RETURN
      END


      SUBROUTINE MODID (ID)
C
C  ADD A CAMERA ID (if needed) TO THE CAMERA ID TABLE
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK24.INC'
C
      K=0
```

```
      DO 1010 I=1,NMAX
           IDD=IDCAM(I)
           IF (IDD.EQ.ID) RETURN
           IF (IDD.EQ.0) K=I
 1010 CONTINUE
      IF (K.NE.0) GO TO 1020
C
C  WRITE MESSAGE "ERROR IN SUBROUTINE MODID":
C
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1040) ID,IDCAM
      STOP
 1020 IDCAM(K)=ID
C
      RETURN
C
 1040 FORMAT (' **** ERROR IN SUBROUTINE MODID ****'/20X,'ADDING VARIABL
     .E ',I10/(10X,'VARIABLES ',6I10))
      END


      SUBROUTINE ROTMAT (PAR,R,PR,PQ,RL)
C
C  EVALUATE ROTATION MATRICES AND THEIR PARTIAL DERIVATIVES
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'SWITCH.INC'
      INCLUDE 'OPTION.INC'
      DIMENSION R(3,3), PR(3,3), PQ(3,2), RL(3,3)
      DIMENSION PAR(1), G(3,3), TEMP(3,3)
      DATA ZERO,ONE/0.0D0,1.0D0/
C
C  FORM BASIC ROTATION MATRIX (PHOTO-TO-OBJECT)
C
      SINA=DSIN(PAR(4))
      COSA=DCOS(PAR(4))
      SINB=DSIN(PAR(5))
      COSB=DCOS(PAR(5))
      SINC=DSIN(PAR(6))
      COSC=DCOS(PAR(6))
      R(1,1)=COSB*COSC
      R(1,2)=COSA*SINC+SINA*SINB*COSC
      R(1,3)=SINA*SINC-COSA*SINB*COSC
      R(2,1)=-COSB*SINC
      R(2,2)=COSA*COSC-SINA*SINB*SINC
      R(2,3)=SINA*COSC+COSA*SINB*SINC
      R(3,1)=SINB
      R(3,2)=-SINA*COSB
      R(3,3)=COSA*COSB
      IF (IATT.EQ.0) GO TO 1020
      DO 1010 I=1,3
           DO 1010 J=I,3
```

```
          IF (I.EQ.J) GO TO 1010
          CON=R(I,J)
          R(I,J)=R(J,I)
          R(J,I)=CON
 1010 CONTINUE
      IF (IS.EQ.0) GO TO 1030
      PR(1,1)=ONE
      PR(1,2)=ZERO
      PR(1,3)=SINB
      PR(2,1)=ZERO
      PR(2,2)=COSA
      PR(2,3)=-SINA*COSB
      PR(3,1)=ZERO
      PR(3,2)=SINA
      PR(3,3)=COSA*COSB
      GO TO 1030
 1020 IF (IS.EQ.0) GO TO 1030
      PR(1,1)=-COSB*COSC
      PR(1,2)=-SINC
      PR(1,3)=ZERO
      PR(2,1)=COSB*SINC
      PR(2,2)=-COSC
      PR(2,3)=ZERO
      PR(3,1)=-SINB
      PR(3,2)=ZERO
      PR(3,3)=-ONE
C
C  FORM LOCAL-TO-GEOCENTRIC MATRIX
C
 1030 CALL COPY (R,RL,9)
      IF (IUNIT.EQ.0) GO TO 1040
      SINA=DSIN(PAR(1))
      COSA=DCOS(PAR(1))
      SINB=DSIN(PAR(2))
      COSB=DCOS(PAR(2))
      G(1,1)=-SINA
      G(1,2)=-COSA*SINB
      G(1,3)=COSA*COSB
      G(2,1)=COSA
      G(2,2)=-SINA*SINB
      G(2,3)=SINA*COSB
      G(3,1)=ZERO
      G(3,2)=COSB
      G(3,3)=SINB
      CALL MPYAB (G,R,TEMP,3,3,3)
      CALL COPY (TEMP,R,9)
      IF (IS.EQ.0) GO TO 1040
      CALL MPYAB (G,PR,TEMP,3,3,3)
      CALL COPY (TEMP,PR,9)
      PQ(1,1)=ZERO
      PQ(1,2)=SINA
      PQ(2,1)=ZERO
      PQ(2,2)=-COSA
      PQ(3,1)=ONE
      PQ(3,2)=ZERO
```

```
C
 1040 RETURN
      END

      SUBROUTINE INVRT (A,N,L,M,N1)
C
C  FIND THE INVERSE OF A MATRIX BY THE GAUSSIAN ELIMINATION METHOD
C
C    A  = array in which the matrix to be inverted is located
C    N  = the second last dimension of A
C    L  = vector of dimension N used by INVERT temporarily
C    M  = vector of dimension N used by INVERT temporarily
C    N1 = order of the submatrix to be inverted
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1), L(1), M(1)
C
C  Initiate the continued product of pivots which will become the
C  Determinant of the matrix and start the main elimination loop
C
      DO 1170 K=1,N1
C
C  Search for the largest element
C
           L(K)=K
           M(K)=K
           KK=K+N*(K-1)
           BIGA=A(KK)
           DO 1020 I=K,N1
                DO 1020 J=K,N1
                IJ=I+N*(J-1)
                IF (DABS(BIGA)-DABS(A(IJ))) 1010,1020,1020
 1010           BIGA=A(IJ)
                L(K)=I
                M(K)=J
 1020      CONTINUE
C
C  A zero largest element means the largest matrix in A is less
C  than N by N
C
           IF (BIGA) 1030,1180,1030
C
C  Interchange rows
C
 1030      J=L(K)
           IF (L(K)-K) 1060,1060,1040
 1040      DO 1050 I=1,N1
                KI=K+N*(I-1)
                HOLD=-A(KI)
                JI=J+N*(I-1)
                A(KI)=A(JI)
 1050           A(JI)=HOLD
 1060      I=M(K)
           IF (M(K)-K) 1090,1090,1070
```

```
 1070       DO 1080 J=1,N1
                 JK=J+N*(K-1)
                 HOLD=-A(JK)
                 JI=J+N*(I-1)
                 A(JK)=A(JI)
 1080            A(JI)=HOLD
C
C  Divide column by minus pivot
C
 1090       DO 1110 I=1,N1
                 IF (I-K) 1100,1110,1100
 1100            IK=I+N*(K-1)
                 A(IK)=A(IK)/(-A(KK))
 1110       CONTINUE
C
C  Reduce matrix
C
            DO 1140 I=1,N1
                 DO 1140 J=1,N1
                 IF (I-K) 1120,1140,1120
 1120            IF (J-K) 1130,1140,1130
 1130            IJ=I+N*(J-1)
                 IK=I+N*(K-1)
                 KJ=K+N*(J-1)
                 A(IJ)=A(IK)*A(KJ)+A(IJ)
 1140       CONTINUE
C
C  Divide row by pivot
C
            DO 1160 J=1,N1
                 IF (J-K) 1150,1160,1150
 1150            KJ=K+N*(J-1)
                 A(KJ)=A(KJ)/A(KK)
 1160       CONTINUE
C
C
            A(KK)=1./A(KK)
 1170 CONTINUE
C
C  Final row and column interchange:
C
       K=N1
 1180 K=K-1
       IF (K) 1250,1250,1190
 1190 I=L(K)
       IF (I-K) 1220,1220,1200
 1200 DO 1210 J=1,N1
            JK=J+N*(K-1)
            HOLD=A(JK)
            JI=J+N*(I-1)
            A(JK)=-A(JI)
 1210       A(JI)=HOLD
 1220 J=M(K)
       IF (J-K) 1180,1180,1230
 1230 DO 1240 I=1,N1
```

```
            KI=K+N*(I-1)
            HOLD=A(KI)
            JI=J+N*(I-1)
            A(KI)=-A(JI)
 1240       A(JI)=HOLD
      GO TO 1180
 1250 CONTINUE
C
      RETURN
      END


      SUBROUTINE LEASTQ (ITAPE,JTAPE,KTAPE,LTAPE)
C
C  PERFORM LEAST SQUARES SOLUTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK21.INC'
      REAL*4           PLATE
      COMMON /WORK23/  SUBM(72),OBJECT(3,4),CORRM(18,ISZ4),PIVOT(3,3),
     .                 EPS(3),TMP(6,3),GXYZ(3),DXYZ(3,3),PLATE(2,ISZ4),
     .                 IDCAM(ISZ4),INRC(3),XDMM(66)
      INCLUDE 'ROTAT.INC'
      INCLUDE 'COEFF.INC'
      INCLUDE 'SWITCH.INC'
      INCLUDE 'OPTION.INC'
      INCLUDE 'UNITVR.INC'
      DIMENSION        SUBMAT(6,6), SUBVEC(6), IA(3), IB(3)
      EQUIVALENCE      (SUBM(1),SUBMAT(1,1)), (TMP(1,1),SUBVEC(1))
      DATA ZERO,ONEM   /0.0D0,-1.0D0/
C
C  INITIALIZATIONS
C
C  ** ITAPE ** LEAST SQUARES POINTERS.
C  ** JTAPE ** OBJECT POINT DATA.
C  ** KTAPE ** OBJECT POINT NORMALS.
C  ** LTAPE ** CAMERA STATION NORMALS.
C
      IS=1
      REWIND ITAPE
      REWIND JTAPE
      REWIND KTAPE
      REWIND LTAPE
      SC=ZERO
      SG=ZERO
      SI=ZERO
C
C  READ SORTED TRIANGULATION DATA
C
 1010 READ (ITAPE) INRC
      N=INRC(2)
C
C  TEST FOR TYPE OF RECORD:
C   If N positive - Object Point data.
```

```
C    If N negative - END Camera Data signal.
C    If N zero - End Of File.
C
      IF (N) 1020,1130,1060
 1020 N = - N
      IF (INRC(3).LT.0) GO TO 1050
C
C   INITIALIZE FOR CAMERA STATION PARAMETERS
C
      CALL FILL (SUBM,36,ZERO)
      DO 1030 I=1,6
           CON=WTMAT(I,N)
           COM=ACCSOL(I,N)
           SUBMAT(I,I)=CON
           SUBVEC(I)=-COM*CON
           SC=SC+COM*COM*CON
 1030 CONTINUE
      I=32769*N
      CALL STSUBM (SUBMAT,I,0)
      CALL STSUBV (SUBVEC,N,0)
C
C   FORM CAMERA STATION ROTATION AND POSITION PARAMETERS
C
      CALL LOCTID (N,ID)
      CALL ROTMAT(PARAM(1,N),R(1,1,ID),PR(1,1,ID),PQ(1,1,ID),RL(1,1,ID))
      IF (IUNIT.EQ.0) GO TO 1040
      CALL PLHXYZ (PARAM(1,N),STATON(1,ID),DSTATN(1,1,ID))
      GO TO 1010
 1040 CALL COPY (PARAM(1,N),STATON(1,ID),3)
      GO TO 1010
 1050 CALL DROP (N,LTAPE)
      GO TO 1010
C
C   READ OBJECT POINT DATA
C
 1060 READ (JTAPE) (IDCAM(K),K=1,N),OBJECT,((PLATE(I,J),I=1,2),J=1,N)
C
C   FORM CONDITION EQUATIONS
C
      CALL FILL (PIVOT,9,ZERO)
      DO 1070 I=1,3
           CON=OBJECT(I,2)
           COM=OBJECT(I,3)
           PIVOT(I,I)=CON
           EPS(I)=-CON*COM
           SG=SG+COM*COM*CON
 1070 CONTINUE
      IF (IUNIT.EQ.0) GO TO 1080
      CALL PLHXYZ (OBJECT,GXYZ,DXYZ)
      GO TO 1090
 1080 CALL COPY (OBJECT,GXYZ,3)
 1090 DO 1110 II=1,N
           ID=IDCAM(II)
           CALL CONEQN (ID,GXYZ,DXYZ,PLATE(1,II),OBJECT(3,1))
           DO 1100 I=1,2
```

```
                CON=VARPLT(I,ID)
                DO 1100 J=1,10
                A(I,J)=CON*A(I,J)
 1100       CONTINUE
            SI=SI+C(1)*C(1)+C(2)*C(2)
            IDD=ID+32768*ID
            CALL MPYATB (B,B,SUBM,6,2,6)
            CALL STSUBM (SUBM,IDD,1)
            CALL MPYATB (B,C,SUBM,6,2,1)
            CALL STSUBV (SUBM,ID,1)
            CALL MPYATB (A,A,SUBM,3,2,3)
            CALL ADDMAT (SUBM,PIVOT,PIVOT,9)
            CALL MPYATB (A,C,SUBM,3,2,1)
            CALL ADDMAT (SUBM,EPS,EPS,3)
            CALL MPYATB (A,B,CORRM(1,II),3,2,6)
 1110 CONTINUE
C
C  ELIMINATE OBJECT POINT COORDINATES
C
      CALL INVRT (PIVOT,3,IA,IB,3)
      WRITE (KTAPE) PIVOT,EPS,N,(IDCAM(I),I=1,N),
     .                          ((CORRM(I,J),I=1,18),J=1,N)
      CALL MPYAB (PIVOT,ONEM,PIVOT,9,1,1)
      DO 1120 I=1,N
            ID1=IDCAM(I)
            CALL MPYATB (CORRM(1,I),PIVOT,TMP,6,3,3)
            CALL MPYAB (TMP,EPS,SUBM,6,3,1)
            CALL STSUBV (SUBM,ID1,1)
            DO 1120 J=I,N
            ID2=ID1+32768*IDCAM(J)
            CALL MPYAB (TMP,CORRM(1,J),SUBM,6,3,6)
            CALL STSUBM (SUBM,ID2,1)
 1120 CONTINUE
      GO TO 1010
C
 1130 SS=SC+SG+SI
      RETURN
      END


      SUBROUTINE STSUBM (REC,IDBLK,IND)
C
C  Accumulate, Initialize, or Extract a 6x6 submatrix C of
C      the normal equations
C
C     IND = 0, Initialize the submatrix.
C     IND = 1, Accumulate to the submatrix.
C     IND =-1, Extract the submatrix.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK22.INC'
      DIMENSION REC(72)
C
C  Decode Camera IDentification
```

```
C
      ID2=IDBLK/32768
      ID1=IDBLK-ID2*32768
C
C  Extract camera position integers
C
      IF (IND.LT.0) GO TO 1010
      CALL MODID (ID1)
      CALL MODID (ID2)
 1010 CALL LOCTID (ID1,I)
      CALL LOCTID (ID2,J)
C
C  Locate block position
C
      L=1
      IF (J.GE.I) GO TO 1020
      K=I
      I=J
      J=K
      L=37
 1020 K=(I+(J*(J-1))/2)*36-35
      IF (IND) 1060,1040,1030
 1030 IF (L.NE.1) CALL TRANSP (REC,REC(37))
      CALL ADDMAT (REC(L),EQN(K),EQN(K),36)
      GO TO 1080
 1040 IF (L.NE.1) GO TO 1050
      CALL COPY (REC,EQN(K),36)
      GO TO 1080
 1050 CALL TRANSP (REC,EQN(K))
      GO TO 1080
 1060 IF (L.NE.1) GO TO 1070
      CALL COPY (EQN(K),REC,36)
      GO TO 1080
 1070 CALL TRANSP (EQN(K),REC)
C
 1080 RETURN
      END


      SUBROUTINE STSUBV (REC,IDBLK,IND)
C
C  Accumulate, Initialize, or Extract a 6x1 subvector  of
C      the normal equation Constant terms
C
C    IND = 0, Initialize the subvector.
C    IND = 1, Accumulate to the subvector.
C    IND =-1, Extract the subvector.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK22.INC'
      DIMENSION       REC(6)
C
      IF (IND.LT.0) GO TO 1010
      CALL MODID (IDBLK)
```

```
 1010 CALL LOCTID (IDBLK,I)
      K=6*I-5
      IF (IND) 1040,1030,1020
 1020 CALL ADDMAT (REC,CONV(K),CONV(K),6)
      GO TO 1050
 1030 CALL COPY (REC,CONV(K),6)
      GO TO 1050
 1040 CALL COPY (CONV(K),REC,6)
C
 1050 RETURN
      END


      SUBROUTINE CONEQN (IDIN,OBJECT,DGROND,PLATE,ELV)
C
C  EVALUATE COLLINEARITY CONDITION EQUATIONS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL*4           PLATE
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK21.INC'
      COMMON /COEFF/  AIM(2,3),EIM(2),BIM(2,6)
      INCLUDE 'ROTAT.INC'
      INCLUDE 'OPTION.INC'
      INCLUDE 'OPTON4.INC'
      DIMENSION       OBJECT(3), DGROND(3,3), PLATE(2), A(2), VG(3),
     .                VC(3), S(3,3), TEMP(2,3), TEMM(2,2)
      DATA S          /9*0.0D0/
C
C  Determine internal position of camera station parameters
C
      CALL LOCTID (IDIN,ID)
C
C  Correct image coordinates for Refraction if called for
C
      IF (IAREFR.EQ.0.OR.IWREFR.EQ.0) CALL REFRCT (PLATE,FOCAL(IDIN),
     .PARAM(3,IDIN),ELV,RL(1,1,ID))
C
C  Compute OBJECT TO CAMERA Vector (Object Space)
C
      CALL SUBMAT (OBJECT,STATON(1,ID),VG,3)
C
C  Compute OBJECT TO CAMERA Vector (Camera Space)
C
      CALL MPYATB (R(1,1,ID),VG,VC,3,3,1)
      A(1)=VC(1)/VC(3)
      A(2)=VC(2)/VC(3)
      C=FOCAL(IDIN)/VC(3)
C
C  Form coefficients of rectangular object coordinates
C
      DO 1010 I=1,2
           CON=A(I)
           DO 1010 J=1,3
           VAL=C*(CON*R(J,3,ID)-R(J,I,ID))
```

```
            AIM(I,J)=VAL
            BIM(I,J)=-VAL
 1010 CONTINUE
C
C  Form constant vector EIM
C
      EIM(1)=C*VC(1)-PLATE(1)
      EIM(2)=C*VC(2)-PLATE(2)
C
C  Form coefficients of differential rotation vector
C
      S(1,2)=-VG(3)
      S(1,3)=VG(2)
      S(2,1)=VG(3)
      S(2,3)=-VG(1)
      S(3,1)=-VG(2)
      S(3,2)=VG(1)
      CALL MPYAB (AIM,S,TEMP,2,3,3)
      CALL MPYAB (TEMP,PR(1,1,ID),BIM(1,4),2,3,3)
      IF (IUNIT.EQ.0) GO TO 1020
C
C Adjust condition equations for Geographic Reference System
C
      CALL MPYAB (TEMP,PQ(1,1,ID),TEMM,2,3,2)
      CALL MPYAB (AIM,DGROND,TEMP,2,3,3)
      CALL COPY (TEMP,AIM,6)
   \  CALL MPYAB (BIM,DSTATN(1,1,ID),TEMP,2,3,3)
      CALL COPY (TEMP,BIM,6)
      CALL ADDMAT (BIM,TEMM,BIM,4)
C
C  Normalize condition equations
C
 1020 RETURN
      END


      SUBROUTINE BACKSL (ITAPE,JTAPE)
C
C  COMPUTE THE BACK SOLUTION FOR THE ELIMINATION PROCESS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK21.INC'
      INCLUDE 'WORK22.INC'
      INCLUDE 'WORK24.INC'
      COMMON /WORK23/ SUBM(6,6),CV(6),CV1(6),XDUM(144),XDUM2(18,ISZ4),
     .                IDUM(3,ISZ4),IDUM2(3)
      INCLUDE 'OPTON2.INC'
C
C  ** ITAPE ** CAMERA STATION NORMALS
C  ** JTAPE ** REVERSED ORDER CAMERA STATION NORMALS
C
      REWIND JTAPE
C
C  COMPUTE SOLUTION OF CAMERA STATIONS
```

```
C
      DO 1040 J=1,NCAM
           BACKSPACE ITAPE
           READ (ITAPE) N,M,ID,IDS,SUBM,CV,
     .                                  ((TMPST(K,L),K=1,36),L=1,M)
           IF (IPROP.NE.0) WRITE (JTAPE) N,M,ID,IDS,SUBM,
     .                                  ((TMPST(K,L),K=1,36),L=1,M)
           IF (N.EQ.0) GO TO 1020
           DO 1010 I=1,N
                IDD=IDS(I)
                CALL MPYATB (TMPST(1,I),SOLUTM(1,IDD),CV1,6,6,1)
                CALL SUBMAT (CV,CV1,CV,6)
 1010      CONTINUE
 1020      CALL MPYAB (SUBM,CV,SOLUTM(1,ID),6,6,1)
           DO 1030 I=1,6
                CON=SOLUTM(I,ID)
                PARAM(I,ID)=PARAM(I,ID)+CON
                ACCSOL(I,ID)=ACCSOL(I,ID)+CON
 1030      CONTINUE
           BACKSPACE ITAPE
 1040 CONTINUE
C
      REWIND ITAPE
      REWIND JTAPE
      RETURN
      END


      SUBROUTINE UPDATG (ITAPE,JTAPE,KTAPE,LTAPE)
C
C  COMPUTE AND UPDATE OBJECT POSITIONS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'WORK21.INC'
      REAL*4          PLATE
      COMMON /WORK23/ OBJECT(3,4),PIVOT(3,3),EPS(3),CORRM(18,ISZ4),
     .                VEC(3),PLATE(2,ISZ4),IDCAM(ISZ4),INRC(3),
     .                XDUM(165)
C
C  ** ITAPE ** LEAST SQUARES POINTERS
C  ** JTAPE ** INPUT OBJECT DATA FILE
C  ** KTAPE ** OUTPUT OBJECT DATA FILE
C  ** LTAPE ** OBJECT POINT NORMALS
C
      REWIND ITAPE
      REWIND JTAPE
      REWIND KTAPE
      REWIND LTAPE
C
 1010 READ (ITAPE) INRC
      N=INRC(2)
      IF (N) 1010,1060,1020
C
C  Compute Object Correction
```

```
C
 1020 READ (JTAPE) (IDCAM(I),I=1,N),OBJECT,((PLATE(I,J),I=1,2),J=1,N)
      READ (LTAPE) PIVOT,EPS,M,(IDCAM(I),I=1,M),
     .                                     ((CORRM(I,J),I=1,18),J=1,M)
      IF (N.EQ.M) GO TO 1030
      CALL CLR
      CALL TOPLFT
      CALL CURDWN (8)
      CALL BEEP
      WRITE (*,1070) N,M
      STOP
 1030 DO 1040 I=1,N
           ID=IDCAM(I)
           CALL MPYAB (CORRM(1,I),SOLUTM(1,ID),VEC,3,6,1)
           CALL SUBMAT (EPS,VEC,EPS,3)
 1040 CONTINUE
      CALL MPYAB (PIVOT,EPS,VEC,3,3,1)
C
C  Update Object Coordinates
C
      IND=INRC(3)
      DO 1050 I=1,3
           CON=VEC(I)
           OBJECT(I,4)=CON
           ICODE=MOD(IND,2)
           IND=IND/2
           IF (ICODE.EQ.0) GO TO 1050
           OBJECT(I,3)=CON+OBJECT(I,3)
           OBJECT(I,1)=CON+OBJECT(I,1)
 1050 CONTINUE
      WRITE (KTAPE) (IDCAM(I),I=1,N),OBJECT,((PLATE(I,J),I=1,2),J=1,N)
      GO TO 1010
 1060 I=JTAPE
      JTAPE=KTAPE
      KTAPE=I
C
      RETURN
C
 1070 FORMAT (' **** ERROR IN UPDATG ****   N = 'I2,'  M = ',I2)
      END


      DOUBLE PRECISION FUNCTION PAKDMS (DMS)
C
C  Pack character field into one word
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*15 DMS
      CHARACTER*1  SIGN
      DIMENSION    FACTOR(2)
      DATA FACTOR  /100.0D0,10000.0D0/
C
C  EXECUTE THE EQUIVALENT OF:
C
C  DECODE (15,1000,DMS) SIGN,IDEG,IMIN,SECS
```

```
C
      READ (DMS,1010) SIGN,IDEG,IMIN,SECS
      CON=IDEG*FACTOR(2)+IMIN*FACTOR(1)+SECS
      IF (SIGN.EQ.'-') CON=-CON
      PAKDMS=CON
      RETURN
 1010 FORMAT (A1,2I3,F8.4)
      END


      SUBROUTINE LSTPLR (ITAPE,JTAPE,KTAPE,LTAPE)
C
C  EVALUATE FINAL OBJECT PARAMETERS & LIST IMAGE RESIDUALS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAMS.INC'
      COMMON /TAPES/ IN,IO,IOS,IDUMM(14)
      INCLUDE 'WORK21.INC'
      REAL*4          PT,RESD
      INCLUDE 'WORK25.INC'
      INCLUDE 'OPTION.INC'
      INCLUDE 'OPTON4.INC'
      INCLUDE 'SWITCH.INC'
      INCLUDE 'CONVCR.INC'
      INCLUDE 'UNITVR.INC'
C
      DIMENSION       OBJEKT(3,4),GXYZ(3),DXYZ(3,3),VEC(3),CV(3),
     .                IDGPT(2,ISZ2),PT(2,ISZ4),IDCAM(ISZ4),RESD(2,ISZ4),
     .                ITEMP(2,ISZ4),IRESD(2,ISZ4),INTG(8)
C
      DATA INTG       /'*0*','*1*','*2*','*3*','*4*','*5*','*6*',' '/
      DATA ZERO       /0.0D0/
      DATA MAXLIN     /54/
C
C  ** ITAPE ** POINTER FILE
C  ** JTAPE ** BLOCKED OBJECT DATA
C  ** KTAPE ** INPUT(OBJECT IDENTIFICATIONS) : OUTPUT(CONTROL RESIDUALS)
C  ** LTAPE ** FINAL OBJECT PARAMETERS
C
      IS=0
      LNCTR=80
      REWIND KTAPE
      READ (KTAPE) N,((IDGPT(I,J),I=1,2),J=1,N)
      REWIND KTAPE
C
C  Evaluate contributions to WSSQ (Weighted Sum of the Squares) of the
C  Camera Parameters
C
      SS=ZERO
      CAMSS=ZERO
      GNDSS=ZERO
      PLTSS=ZERO
      DO 1010 I=1,NCAM
           DO 1010 J=1,6
           CON=ACCSOL(J,I)
```

```
            CAMSS=CAMSS+WTMAT(J,I)*CON*CON
 1010 CONTINUE
C
C  Initialize internal Camera IDentifications
C
      CALL INITID
      REWIND ITAPE
      REWIND JTAPE
      REWIND LTAPE
 1020 READ (ITAPE) ID,NP,IND
      IF (NP) 1030,1150,1060
 1030 NP=-NP
      IF (IND.LT.0) GO TO 1050
      CALL MODID (NP)
      CALL LOCTID (NP,ID)
      CALL ROTMAT (PARAM(1,NP),R(1,1,ID),DXYZ,DXYZ,RL(1,1,ID))
      IF (IUNIT.EQ.0) GO TO 1040
      CALL PLHXYZ (PARAM(1,NP),STATON(1,ID),DXYZ)
      GO TO 1020
 1040 CALL COPY (PARAM(1,NP),STATON(1,ID),3)
      GO TO 1020
 1050 CALL DROPID (NP)
      GO TO 1020
 1060 READ (JTAPE) (IDCAM(I),I=1,NP),OBJEKT,((PT(I,J),I=1,2),J=1,NP)
C
C  Final modification of object parameters
C
      COM=ZERO
      INDD=IND
      DO 1080 I=1,3
            ICODE=MOD(INDD,2)
            INDD=INDD/2
            IF (ICODE.NE.0) GO TO 1070
            CON=OBJEKT(I,4)
            OBJEKT(I,1)=OBJEKT(I,1)+CON
            OBJEKT(I,3)=OBJEKT(I,3)+CON
 1070       COM=COM+OBJEKT(I,2)*OBJEKT(I,3)**2
 1080 CONTINUE
      GNDSS=GNDSS+COM
      ID1=IDGPT(1,ID)
      ID2=IDGPT(2,ID)
      WRITE (LTAPE) ID1,ID2,IND,OBJEKT
      IF (IND.LT.7) WRITE (KTAPE) ID1,ID2,IND,(OBJEKT(I,4),I=1,3)
C
C  Estimate plate residuals
C
      IF (IUNIT.EQ.0) GO TO 1090
      CALL PLHXYZ (OBJEKT,GXYZ,DXYZ)
      GO TO 1100
 1090 CALL COPY (OBJEKT,GXYZ,3)
 1100 DO 1110 II=1,NP
            IDC=IDCAM(II)
            CALL LOCTID (IDC,ID)
            CALL SUBMAT (GXYZ,STATON(1,ID),VEC,3)
            CALL MPYATB (R(1,1,ID),VEC,CV,3,3,1)
```

```
            CON=FOCAL(IDC)/CV(3)
            IF (IAREFR .EQ. 0 .OR. IWREFR .EQ. 0) CALL REFRCT (PT(1,II),
     .                                                 FOCAL(IDC),
     .                                                 PARAM(3,IDC),
     .                                                 OBJEKT(3,1),
     .                                                 RL(1,1,ID))
            RESX=CON*CV(1)-PT(1,II)
            RESY=CON*CV(2)-PT(2,II)
            RESD(1,II)=RESX
            RESD(2,II)=RESY
            CON=RESX*VARPLT(1,IDC)
            COM=RESY*VARPLT(2,IDC)
            PLTSS=PLTSS+CON*CON+COM*COM
 1110 CONTINUE
      IF (IRESA.LT.0) GO TO 1020
C
C  Set Missing Control Component Indicators (*0* , *3*, etc.)
C  for Plate Residuals
C
      MISS=INTG(IND+1)
C
C  Identify Image Point (PLATE) Residuals to be listed
C
      NR=0
      DO 1120 I=1,NP
           IDC=IDCAM(I)
           IRESX=1000.0*RESD(1,I)
           IRESY=1000.0*RESD(2,I)
           IF (ABS(IRESX).LT.IRESA.AND.ABS(IRESY).LT.IRESA) GO TO 1120
           NR=NR+1
           IDT=IDCAM(NR)
           IDCAM(NR)=IDC
           IDCAM(I)=IDT
           IRESD(1,NR)=IRESX
           IRESD(2,NR)=IRESY
 1120 CONTINUE
      DO 1130 I=1,NP
           IDC=IDCAM(I)
           ITEMP(1,I)=IFOTO(1,IDC)
           ITEMP(2,I)=IFOTO(2,IDC)
 1130 CONTINUE
C
C  TEST FOR LISTING TITLE PAGE.
C
      IF (LNCTR.LE.MAXLIN) GO TO 1140
      CALL NEWPAG
      WRITE (IO,1170)
      WRITE (IOS,2170)
      LNCTR=5
 1140 IF (NR.EQ.0) GO TO 1020
      LNCTR=LNCTR+1
C
C  List the Point ID, Missing Component Indicator & Photo Numbers
C
      WRITE (IO,1180) ID1,ID2,MISS,((ITEMP(I,J),I=1,2),J=1,NP)
```

```
      WRITE (IOS,2180) ID1,ID2,MISS,((ITEMP(I,J),I=1,2),J=1,NP)
      LNCTR=LNCTR+2
C   Write X-Parallax Residuals for each Photo (12I9 Format)
      WRITE (IO,1190) (IRESD(1,I),I=1,NR)
      WRITE (IOS,2190) (IRESD(1,I),I=1,NR)
C   Write Y-Parallax Residuals for each Photo (12I9 Format)
      WRITE (IO,1190) (IRESD(2,I),I=1,NR)
      WRITE (IOS,2190) (IRESD(2,I),I=1,NR)
C   Skip line
      WRITE (IO,1200)
      WRITE (IOS,1200)
      LNCTR=LNCTR+2
      GO TO 1020
 1150 IF (LNCTR.LE.MAXLIN) GO TO 1160
      CALL NEWPAG
 1160 CONTINUE
C
C  WRITE WEIGHTED SUM OF SQUARES AND THE MAJOR CONTRIBUTORS
C
      SS=CAMSS+GNDSS+PLTSS
      WRITE (IO,1210) CAMSS,GNDSS,PLTSS,SS,IDFREE
      WRITE (*,1220) CAMSS,GNDSS,PLTSS,SS,IDFREE
      WRITE (IOS,1220) CAMSS,GNDSS,PLTSS,SS,IDFREE
C
      VAR2=SS/IDFREE
      VAR=DSQRT(VAR2)
      WRITE (IO,1230) VAR2,VAR
      WRITE (*,1240) VAR2
      WRITE (IOS,1240) VAR2
C
C  SET SS TO VAR2
C
      SS=VAR2
C
      REWIND JTAPE
      REWIND KTAPE
      RETURN
C
 1170 FORMAT (31X,'T R I A N G U L A T E D   I M A G E   P O I N T S
     .R E S I D U A L S'//58X,'(in micrometers)'//)
C
C  Note that the following group of FORMAT statements are for listing
C  Plate Residuals for up to twelve (12) intersections per point:
C
 1180 FORMAT (1X,2A4,1X,A3,1X,12(1X,2A4))
 1190 FORMAT (14X,12I9)
 1200 FORMAT (/)
 1210 FORMAT (/41X,'WEIGHTED SUM OF SQUARES (CAMERA) = ',F15.1/41X,'WEIG
     .HTED SUM OF SQUARES (OBJECT) = ',F15.1/41X,'WEIGHTED SUM OF SQUARE
     .S (PLATES) = ',F15.1//41X,'WEIGHTED SUM OF SQUARES (TOTAL)  = ',
     .F15.1/41X,'DEGREES OF FREEDOM............  = ',6X,I9)
C
 2170 FORMAT (4X,'T R I A N G U L A T E D   I M A G E   P O I N T S
     .R E S I D U A L S'//31X,'(in micrometers)'//)
 2180 FORMAT (1X,2A4,1X,A3,1X,7(1X,2A4))
```

```
 2190 FORMAT (14X,7I9)
 1220 FORMAT (/14X,'Weighted Sum of Squares (Camera) = ',F15.1/14X,'Weig
     .hted Sum of Squares (Object) = ',F15.1/14X,'Weighted Sum of Square
     .s (Plates) = ',F15.1//14X,'Weighted Sum of Squares (Total)  = ',
     .F15.1/14X,'Degrees of Freedom.............  = ',6X,I9)
 1230 FORMAT (//47X,'a posteriori Estimates for Unit Weight'//53X,
     .                        'Variance = ',F15.3/53X,'St. Dev. = ',F15.3)
 1240 FORMAT (///14X,'a posteriori Variance of Unit Weight = ',F15.3)
      END
C
      SUBROUTINE REFRCT (PLATE,FOCAL,BH,SH,RL)
C
C   SUBROUTINE TO CORRECT IMAGE COORDINATES
C   FOR ATMOSPHERIC AND WATER REFRACTION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'OPTON4.INC'
      REAL*4          PLATE(2),FOCAL
      DIMENSION       RL(3,3), P(3), T(3)
      DATA ZERO,ONE   /0.0D0,1.0D0/
C
C   Compute Local Vertical Image Coordinates
C
      P(1)=PLATE(1)
      P(2)=PLATE(2)
      P(3)=FOCAL
    \ CALL MPYAB (RL,P,T,3,3,1)
      TP=T(1)**2+T(2)**2
      TE=T(3)**2
C
C   Evaluate Atmospheric Refraction Constant
c
      IF (IAREFR.EQ.0) THEN
           C1=13.0D-9*(BH-SH)*(ONE-2.0D-5*(BH+BH+SH))
      ELSE
           C1=ZERO
      END IF
C
C   Evaluate Water Refraction Constant
C
      IF (IWREFR.EQ.0.AND.WLEVEL.GT.SH) THEN
           TANSQ=TP/TE
           BWH=BH-WLEVEL
           SWH=SH-WLEVEL
           WH=SWH/SQRT(CNW+(CNW-ONE)*TANSQ)
           C2=(((BWH-SWH)/(BWH-WH))-ONE)/(ONE+TANSQ)
      ELSE
           C2=ZERO
      END IF
      C=C1+C2
C
C   Compute Corrected IMAGE Coordinates in Local Vertical System
C
      C=ONE-C*(TP+TE)/TE
      P(1)=C*P(1)
```

```
      P(2)=C*P(2)
C
C   Compute Corrected IMAGE Coordinates
C
      CALL MPYATB (RL,P,T,3,3,1)
      C=FOCAL/P(3)
      PLATE(1)=C*P(1)
      PLATE(2)=C*P(2)
C
      RETURN
      END


      SUBROUTINE PERROR (ITAPE,JTAPE,KTAPE,LTAPE,MTAPE,NTAPE)
C
C   PERFORM ERROR PROPAGATION (GEOMETRIC DILUTION OF PRECISION [GDOP])
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'PARAMS.INC'
      COMMON /WORK22/ AREA(ISZ8),CONV(ISZ9),WORKC(36,ISZ7),PIVOTC(36)
      INCLUDE 'WORK24.INC'
      INCLUDE 'GPCTRS.INC'
      DIMENSION       OBJECT(3), TEMP(36), SUBM(72), SUBV(3)
      DIMENSION       ZEROM(36), IDP(2), WORKP(18,ISZ4), PIVOTP(9)
      EQUIVALENCE     (WORKC(1,1),WORKP(1,1)), (PIVOTC(1),PIVOTP(1)),
                      (SUBM(1),SUBV(1))
      DATA ZEROM      /36*0.0D0/
      DATA IE1        /ISZ8/
C
C   ** ITAPE ** POINTERS
C   ** JTAPE ** OBJECT POINTS NORMALS (DIRECT)
C   ** KTAPE ** CAMERA PARAMETERS NORMALS (REVERSED)
C   ** LTAPE ** FINAL OBJECT PARAMETERS (WITHOUT COVARIANCES)
C   ** MTAPE ** FINAL OBJECT PARAMETERS (WITH COVARIANCES)
C   ** NTAPE ** OUTPUT CAMERA COVARIANCES
C
C   POSITION DATA SETS
C
      REWIND KTAPE
      REWIND MTAPE
      REWIND NTAPE
      BACKSPACE ITAPE
C
C   INITIALIZE NORMALS
C
      DO 1010 I=1,NMAX
           IDCAM(I)=0
 1010 CONTINUE
      DO 1020 I=1,IE1
           AREA(I)=0.0D0
 1020 CONTINUE
C
C   READ AUTORAY POINTERS
C
      DO 1160 II=1,NIND
```

```
          BACKSPACE ITAPE
          READ (ITAPE) ID,NP,IND
          IF (NP) 1030,1150,1110
 1030     ID=-NP
          IF (IND.LT.0) GO TO 1060
C
C  CAMERA STATION ELIMINATION RECORD
C
          DO 1050 I=1,NMAX
               J=IDCAM(I)
               IF (J.EQ.0) GO TO 1050
               IDBLK=ID+32768*J
               IF (J.NE.ID) GO TO 1040
C
C  EXTRACT AND STORE COVARIANCE MATRIX FOR CAMERA STATION ID
C
               K=I
               CALL STSUBM (SUBM,IDBLK,-1)
               WRITE (NTAPE) ID,(SUBM(N),N=1,36)
C
C  ELIMINATE CORRELATION MATRICES FOR CAMERA STATION ID
C
 1040          CALL STSUBM (ZEROM,IDBLK,0)
 1050     CONTINUE
C
C  ELIMINATE CAMERA STATION ID FROM IDCAM TABLE
C
          IDCAM(K)=0
          GO TO 1150
C
C  CAMERA STATION ADDITION RECORD
C
 1060     READ (KTAPE) N,M,K,IDS,PIVOTC,((WORKC(I,J),I=1,36),J=1,M)
          IF (N.EQ.0) GO TO 1100
          DO 1070 I=1,N
               CALL MPYABT (PIVOTC,WORKC(1,I),SUBM,6,6,6)
               CALL COPY (SUBM,WORKC(1,I),36)
 1070     CONTINUE
C
C  FORM CORRELATION AND COVARIANCE SUBMATRICES FOR CAMERA STATION ID
C
          DO 1090 I=1,N
               CALL COPY (ZEROM,TEMP,36)
               K=32768*IDS(I)
               DO 1080 J=1,N
                    IDBLK=K+IDS(J)
                    CALL STSUBM (SUBM,IDBLK,-1)
                    CALL MPYAB (WORKC(1,J),SUBM,SUBM(37),6,6,6)
                    CALL SUBMAT (TEMP,SUBM(37),TEMP,36)
 1080          CONTINUE
               IDBLK=ID+K
               CALL STSUBM (TEMP,IDBLK,0)
               CALL MPYABT (TEMP,WORKC(1,I),SUBM,6,6,6)
               CALL SUBMAT (PIVOTC,SUBM,PIVOTC,36)
 1090     CONTINUE
```

```
 1100      IDBLK=ID+32768*ID
           CALL STSUBM (PIVOTC,IDBLK,0)
           GO TO 1150
C
C  OBJECT POINT RECORD
C
 1110      BACKSPACE JTAPE
           BACKSPACE LTAPE
           READ (LTAPE) IDP,INDX,OBJECT,PIVOTP
           READ (JTAPE) PIVOTP,SUBV,M,(IDS(I),I=1,M),
                                   ((WORKP(I,J),I=1,18),J=1,M)
C
C  FORM COVARIANCE MATRIX OF OBJECT POINT
C
           DO 1120 I=1,NP
               CALL MPYAB (PIVOTP,WORKP(1,I),SUBM,3,3,6)
               CALL COPY (SUBM,WORKP(1,I),18)
 1120      CONTINUE
           DO 1140 I=1,NP
               CALL COPY (ZEROM,TEMP,18)
               K=32768*IDS(I)
               DO 1130 J=1,NP
                   IDBLK=K+IDS(J)
                   CALL STSUBM (SUBM,IDBLK,-1)
                   CALL MPYAB (WORKP(1,J),SUBM,SUBM(37),3,6,6)
                   CALL SUBMAT (TEMP,SUBM(37),TEMP,18)
 1130          CONTINUE
               CALL MPYABT (TEMP,WORKP(1,I),SUBM,3,6,3)
               CALL SUBMAT (PIVOTP,SUBM,PIVOTP,9)
 1140      CONTINUE
           WRITE (MTAPE) IDP,INDX,OBJECT,PIVOTP
           BACKSPACE JTAPE
           BACKSPACE LTAPE
 1150      BACKSPACE ITAPE
 1160 CONTINUE
C
      RETURN
      END


      SUBROUTINE PLHXYZ (PLH,XYZ,DPLH)
C
C  TRANSFORM COORDINATES & THEIR PARTIALS FROM GEOGRAPHIC TO GEOCENTRIC
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'EARTHD.INC'
      INCLUDE 'SWITCH.INC'
      DIMENSION      PLH(3), XYZ(3), DPLH(3,3)
C
C  Compute Geocentric Coordinates
C
      H=PLH(3)
      ESQ=1.0D0-(SPHRD(2)/SPHRD(1))**2
      SINLA=DSIN(PLH(1))
      COSLA=DCOS(PLH(1))
```

```
      SINFI=DSIN(PLH(2))
      COSFI=DCOS(PLH(2))
      GAMMA=DSQRT(1.0D0-ESQ*SINFI**2)
      CONST=SPHRD(1)/GAMMA
      XYZ(1)=COSFI*COSLA*(CONST+H)
      XYZ(2)=COSFI*SINLA*(CONST+H)
      CONST=H+CONST*(1.0D0-ESQ)
      XYZ(3)=SINFI*CONST
      IF (IS.EQ.0) GO TO 1010
C
C   Compute Matrix of Partials of Geocentric Coordinates
C   with respect to the Geographic Coordinate System
C
      CONST=(CONST-H*ESQ*SINFI**2)/GAMMA**2
      DPLH(1,1)=-XYZ(2)
      DPLH(2,1)=XYZ(1)
      DPLH(3,1)=0.0D0
      DPLH(1,2)=-COSLA*CONST*SINFI
      DPLH(2,2)=-SINLA*CONST*SINFI
      DPLH(3,2)=COSFI*CONST
      DPLH(1,3)=COSFI*COSLA
      DPLH(2,3)=COSFI*SINLA
      DPLH(3,3)=SINFI
C
 1010 RETURN
      END

      SUBROUTINE XYZPLH (XYZ,FLH)
C
C   TRANSFORM COORDINATES FROM GEOCENTRIC TO GEOGRAPHIC
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'EARTHD.INC'
      DIMENSION        XYZ(3), FLH(3)
      DATA PI,PI2      /3.14159265D0,1.570796325D0/
      a = SPHRD (1)
      b = SPHRD (2)
C
C   COMPUTE LONGITUDE
C
      X=XYZ(1)
      Y=XYZ(2)
      Z=XYZ(3)
      CON=0.0D0
      IF (X) 1050,1010,1060
 1010 IF (Y) 1020,1030,1040
 1020 FLH(1)=-PI2
      GO TO 1070
 1030 FLH(1)=0.0D0
      FLH(2)=PI2
      IF (Z.LE.0.0D0) FLH(2)=-PI2
      FLH(3)=DABS(Z)-B
      GO TO 1100
 1040 FLH(1)=PI2
```

```
      GO TO 1070
 1050 CON=PI
      IF (Y.LT.0.0D0) CON=-PI
 1060 FLH(1)=DATAN(Y/X)+CON
C
C  COMPUTE LATITUDE
C
 1070 E2=1.0D0-(B/A)**2
      T1=E2*Z
      DO 1080 I=1,10
           ZP=T1+Z
           SI=ZP/DSQRT(X**2+Y**2+ZP**2)
           CON=DSQRT(1.0D0-E2*SI**2)
           T2=(A*E2*SI)/CON
           IF (DABS(T1-T2).LE.0.005D0) GO TO 1090
           T1=T2
 1080 CONTINUE
      WRITE (*,'(/13H ERROR XYZPLH)')
 1090 RS=X**2+Y**2
      ZP=Z+T2
      FLH(2)=DATAN(ZP/DSQRT(RS))
      T1=A/CON
      FLH(3)=DSQRT(RS+ZP**2)-T1
 1100 RETURN
      END


      SUBROUTINE COPY (A,B,N)
C
C THIS SUBROUTINE COPIES THE FIRST N ELEMENTS OF ARRAY A INTO ARRAY B.
C
C SPECIFICATIONS.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1), B(1)
C
C COPY ARRAY A TO B.
C COPY
      DO 1010 I=1,N
           B(I)=A(I)
 1010 CONTINUE
C
      RETURN
      END
```

# PC Giant

## Source Code

## File Name: 3.FOR (Output)

## 14 June 1990

```
      SUBROUTINE PHASE3
C
C  MAIN CALLING PROGRAM FOR DATA OUTPUT PHASE
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'TAPES.INC'
      INCLUDE 'PARAMS.INC'
      COMMON /WORK31/ PARAM(6,ISZ1),SPCOV(3,3,ISZ1),SACOV(3,3,ISZ1),
     .                IFOTO(2,ISZ1),NCAM
      INCLUDE 'OPTON2.INC'
C
C  READ ADJUSTED CAMERA STATION PARAMETERS
C
      REWIND ITAPE2
      READ (ITAPE2) NCAM,((PARAM(I,J),I=1,6),J=1,NCAM),
     .                 ((IFOTO(I,J),I=1,2),J=1,NCAM)
      REWIND ITAPE2
C
C  Sort triangulated object coordinates if desired (ISORT=0),
C  List triangulated object coordinates,
C  give statistical summary of changes to input object control
C     if it exists (NCNTRL=1).
C
      IF (ISORT.EQ.0) CALL SRTGPS (ITAPE3,ITAPE4,ITAPE7)
      CALL LSTPNH (ITAPE3,ITAPE5)
      IF (NCNTRL .NE. 0) CALL LSTGRS (ITAPE6)
      IF (IANTH.NE.0)CALL ANTHRO
      RETURN
      END


      SUBROUTINE SRTGPS (ITAPE,JTAPE,KTAPE)
C
C  THIS PROGRAM SORTS THE OBJECT POINTS IN ASCENDING
C  IDENTIFICATION ORDER.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*4      CDTAB(3,613),CDTAP(3),MIN1,MIN2
      COMMON /TAPES/   IN,IO,IOS,IDUM(14)
      COMMON /WORK21/  GPTAB(12,613),IDTAB(3,613)
      INCLUDE 'GPCTRS.INC'
      DIMENSION        IDTAP(3), GPTAP(12)
      EQUIVALENCE      (IDTAB(1,1),CDTAB(1,1)), (IDTAP(1),CDTAP(1))
      DATA IFULL       /613/
      DATA INDX        /1073741825/
C
C  ** ITAPE ** OBJECT DATA
C  ** JTAPE ** SCRATCH DATA SET
C  ** KTAPE ** SCRATCH DATA SET
C
C  INITIALIZATION:
C
      N=NGPS
      IPASS=0
      REWIND ITAPE
```

```
 1010 REWIND JTAPE
      REWIND KTAPE
      MTBL=0
      IPASS=IPASS+1
C
C  READ A OBJECT POINT.
C
 1020 MTBL=MTBL+1
      READ  (ITAPE)  (IDTAB(I,MTBL),I=1,3),(GPTAB(I,MTBL),I=1,12)
      N=N-1
C
C  CHECK FOR FULL TABLE OR LAST GP (Object Point).
C
      IF  (MTBL.NE.IFULL.AND.N.NE.0) GO TO 1020
C
C  SORT OBJECT POINTS IN ASCENDING IDENT ORDER.
C
      DO 1070 I=1,MTBL
           IF  (I.EQ.MTBL) GO TO 1070
           MIN1=CDTAB(1,I)
           MIN2=CDTAB(2,I)
           IDX=0
           K=I+1
           DO 1040 J=K,MTBL
                IF  (CDTAB(1,J).LT.MIN1) THEN
                     GO TO 1030
                ELSE IF  (CDTAB(1,J).GT.MIN1) THEN
                     GO TO 1040
                END IF
                IF  (CDTAB(2,J).GE.MIN2) GO TO 1040
 1030           MIN1=CDTAB(1,J)
                MIN2=CDTAB(2,J)
                IDX=J
 1040      CONTINUE
           IF  (IDX.EQ.0)  GO TO 1070
           DO 1050 K=1,3
                MIN1=CDTAB(K,I)
                CDTAB(K,I)=CDTAB(K,IDX)
 1050           CDTAB(K,IDX)=MIN1
           DO 1060 K=1,12
                CON=GPTAB(K,I)
                GPTAB(K,I)=GPTAB(K,IDX)
 1060           GPTAB(K,IDX)=CON
 1070 CONTINUE
C
C  CHECK FOR FIRST DATA PASS.
C
      K=1
      IF  (IPASS.EQ.1)  GO TO 1130
C
C  READ PREVIOUS GP FROM TAPE.
C
 1080 READ  (JTAPE)  IDTAP,GPTAP
      IF  (IDTAP(3).EQ.INDX)  GO TO 1130
C
```

```
C  Check the idents of the two Ground Points.
C
 1090 IF (CDTAB(1,K).LT.CDTAP(1)) THEN
           GO TO 1100
      ELSE IF (CDTAB(1,K).GT.CDTAP(1)) THEN
           GO TO 1120
      END IF
      IF (CDTAB(2,K).GT.CDTAP(2)) GO TO 1120
C
C  IDENT OF GP IN MEMORY IS Less Than IDENT OF GP ON TAPE.
C
 1100 WRITE (KTAPE) (IDTAB(I,K),I=1,3),(GPTAB(I,K),I=1,12)
      K=K+1
      IF (K.LE.MTBL) GO TO 1090
C
C  MEMORY EXHAUSTED. WRITE TAPE Ground Points until tape is exhausted.
C
 1110 WRITE (KTAPE) IDTAP,GPTAP
      READ (JTAPE) IDTAP,GPTAP
      IF (IDTAP(3).EQ.INDX) GO TO 1140
      GO TO 1110
C
C  IDENT OF GP ON TAPE IS Less Than IDENT OF GP IN MEMORY.
C
 1120 WRITE (KTAPE) IDTAP,GPTAP
      GO TO 1080
C
C  TAPE EXHAUSTED. Write MEMORY Ground Points until memory is exhausted.
C
 1130 WRITE (KTAPE) (IDTAB(I,K),I=1,3),(GPTAB(I,K),I=1,12)
      K=K+1
      IF (K.LE.MTBL) GO TO 1130
C
C  WRITE DATA SENTINEL & ALTERNATE TAPES FOR NEXT DATA PASS.
C
 1140 IDTAP(3)=INDX
      WRITE (KTAPE) IDTAP,GPTAP
      I=JTAPE
      JTAPE=KTAPE
      KTAPE=I
C
C  CHECK FOR FINAL END OF OBJECT POINTS
C
      IF (N.NE.0) GO TO 1010
      I=ITAPE
      ITAPE=JTAPE
      JTAPE=I
      REWIND ITAPE
      REWIND JTAPE
      REWIND KTAPE
C
      RETURN
      END
```

```
      SUBROUTINE LSTPNH (ITAPE,JTAPE)
C
C   THIS PROGRAM LISTS AND/OR PUNCHES THE TRIANGULATED RESULTS
C   OF THE GIANT BLOCK ADJUSTMENT PROGRAM.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      LOGICAL*4         BTEST
      CHARACTER*15      IDMSS(3),IDMS1,IDMS2,IDMS3,IDMS4,IDMS5,CH(3)
      CHARACTER*17      IGRPH(0:1)
      COMMON /TAPES/    IN,IO,IOS,IP1,IP2,IDUM(12)
      INCLUDE 'EARTHD.INC'
      INCLUDE 'PARAMS.INC'
      COMMON /WORK31/ PARAM(6,ISZ1),SPCOV(3,3,ISZ1),SACOV(3,3,ISZ1),
     .                IDPHO(2,ISZ1),NCAM
      INCLUDE 'OPTION.INC'
      INCLUDE 'OPTON2.INC'
      INCLUDE 'GPCTRS.INC'
      INCLUDE 'UNITVR.INC'
      COMMON /ANTHR/P(7, 3)
      CHARACTER*19 IOFM1, IOFM2
      DATA IOFM1/'(2A4,3F12.3,3G10.4)'/
      DATA IOFM2/'(2A4,3F12.3,3G10.4)'/
C
      DIMENSION    STATN(6), COVARS(6,6), OBJECT(3), GPCOV(3,3)
      DIMENSION    INDTYP(8),    SSCVP(3), SSCVA(3), NOSS(3)
      DIMENSION    SPC(3, 3), EVEC(3, 3), EVAL(3), EVX(3)
      EQUIVALENCE  (IDMSS(1),IDMS1), (IDMSS(2),IDMS2),
     .             (IDMSS(3),IDMS3)
      DATA IGRPH   /'(Photo to Object)','(Object to Photo)'/
      DATA INDTYP  /'*0*','*1*','*2*','*3*','*4*','*5*','*6*','    '/
      DATA IEND    /'****'/
      DATA MAXLIN  /56/
      DATA ZERO    /0.0D0/
C
C   ** ITAPE ** OBJECT DATA
C   ** JTAPE ** CAMERA PARAMETERS COVARIANCES
C
C   BEGIN TO PROCESS THE CAMERA STATIONS AND
C      INITIALIZE FOR CAMERA STATIONS.
C
      LNCTR=80
      NCNTRL=0
      IF(IPNST.EQ.0) OPEN (UNIT=IP1,STATUS='UNKNOWN',FILE='cam.out')
      IF(IPNGP.EQ.0) OPEN (UNIT=IP2,STATUS='UNKNOWN',FILE='obj.out')
      REWIND ITAPE
      REWIND JTAPE
C
C   CHECK TO LOAD THE CAMERA STATIONS' COVARIANCES
C
      IF (IWGHT.EQ.2) SS=1.0D0
      IF (IPROP.EQ.0) GO TO 1050
      DO 1010 I=1,3
           SSCVP(I)=ZERO
           SSCVA(I)=ZERO
 1010 CONTINUE
```

```
      DO 1030 II=1,NCAM
           READ (JTAPE) I,COVARS
           DO 1020 J=1,3
                DO 1020 K=1,3
                SPCOV(J,K,I)=COVARS(J,K)*SS
                SACOV(J,K,I)=COVARS(J+3,K+3)*SS
                IF (J.EQ.K) THEN
                     SSCVP(J)=SSCVP(J)+SPCOV(J,K,I)
                     SSCVA(J)=SSCVA(J)+SACOV(J,K,I)
                END IF
 1020      CONTINUE
 1030 CONTINUE
      DO 1040 I=1,3
           SSCVP(I)=SQRT(SSCVP(I)/FLOAT(NCAM))
           SSCVA(I)=SQRT(SSCVA(I)/FLOAT(NCAM))
 1040 CONTINUE
 1050 IF (ILTST.NE.0.AND.IPNST.NE.0) GO TO 1250
      NSTA=0
C
C  CHECK OPTION FOR LISTING CAMERA STATIONS
C
 1060 IF (ILTST.NE.0) GO TO 1090
C
C  CHECK TO LIST THE PAGE HEADING
C
      IF (LNCTR.LT.MAXLIN) GO TO 1100
      CALL NEWPAG
      WRITE (IO,1430)
      WRITE (IOS,2430) IGRPH(IATT)
      LNCTR=4
      IF (IPROP.NE.0) GO TO 1080
      WRITE (IO,1440) IGRPH(IATT)
      WRITE (IOS,2440)
      GO TO 1100
 1080 IF(IEIGEN.NE.0) THEN
          WRITE (IO,1450) IGRPH(IATT)
          WRITE (IOS,2450)
      ENDIF
      IF(IEIGEN.EQ.0) THEN
          WRITE (IO,1455) IGRPH(IATT)
          WRITE (IOS,2455)
      ENDIF
      GO TO 1100
C
C  CHECK OPTION FOR PUNCHING CAMERA STATIONS
C
 1090 IF (IPNST.NE.0) GO TO 1220
C
C  PICK UP IDENT AND PARAMETERS OF CAMERA STATION,
C  CONVERT ATTITUDE IF NEED BE.
C
 1100 NSTA=NSTA+1
      ID1=IDPHO(1,NSTA)
      ID2=IDPHO(2,NSTA)
      DO 1110 I=1,6
```

```
 1110      STATN(I)=PARAM(I,NSTA)
      DO 1120 I=1,3
           J=I+3
           CALL RADDEG (STATN(J),IDMSS(I))
 1120 CONTINUE
C
C  CHECK THE UNITS OF THE STATION PARAMETERS
C
      IF (IUNIT.NE.0) GO TO 1170
C
C  LOCAL UNITS; CHECK TO LIST THE CAMERA STATION
C
      IF (ILTST.NE.0) GO TO 1150
      IF (IPROP.NE.0) GO TO 1130
      WRITE (IO,1460)         STATN(1),IDMS1
      WRITE (IO,1470) ID1,ID2,STATN(2),IDMS2
      WRITE (IO,1480)         STATN(3),IDMS3
      WRITE (IOS,2460)         STATN(1),IDMS1
      WRITE (IOS,2470) ID1,ID2,STATN(2),IDMS2
      WRITE (IOS,2480)         STATN(3),IDMS3
      GO TO 1140
 1130 IF(IEIGEN.NE.0)GO TO 1138
      DO 1131 I=1, 3
      DO 1131 J=1, 3
 1131 SPC(I, J)=SPCOV(I, J, NSTA)
      CALL TRED2(3, 3, SPC, EVAL, EVX, EVEC)
      CALL TQL2 (3, 3, EVAL, EVX, EVEC, IERR)
      DO 11315 I=1, 3
11315 CALL RADDEG(DSQRT(SACOV(I,I,NSTA)),CH(I))
      WRITE (IO,1132)         STATN(1),(EVEC(I,3),I=1,3),DSQRT(EVAL(3)),
     .                           IDMS1,CH(1)
      WRITE (IOS,2132)        STATN(1),(EVEC(I,3),I=1,3),DSQRT(EVAL(3))
      WRITE (IO,1134) ID1,ID2,STATN(2),(EVEC(I,2),I=1,3),DSQRT(EVAL(2)),
     .                           IDMS2,CH(2)
      WRITE (IOS,2134)ID1,ID2,STATN(2),(EVEC(I,2),I=1,3),DSQRT(EVAL(2))
      WRITE (IO,1136)         STATN(3),(EVEC(I,1),I=1,3),DSQRT(EVAL(1)),
     .                           IDMS3,CH(3)
      WRITE (IOS,2136)        STATN(3),(EVEC(I,1),I=1,3),DSQRT(EVAL(1)),
     .                           IDMS1,CH(1),IDMS2,CH(2),IDMS3,CH(3)
      GO TO 1140
 1138 WRITE (IO,1490)         STATN(1),(SPCOV(1,I,NSTA),I=1,3),
     .                           IDMS1,(SACOV(1,I,NSTA),I=1,3)
      WRITE (IO,1500) ID1,ID2,STATN(2),(SPCOV(2,I,NSTA),I=1,3),
     .                           IDMS2,(SACOV(2,I,NSTA),I=1,3)
      WRITE (IO,1510)         STATN(3),(SPCOV(3,I,NSTA),I=1,3),
     .                           IDMS3,(SACOV(3,I,NSTA),I=1,3)
      WRITE (IOS,2490)        STATN(1),(SPCOV(1,I,NSTA),I=1,3)
      WRITE (IOS,2500)ID1,ID2,STATN(2),(SPCOV(2,I,NSTA),I=1,3)
      WRITE (IOS,2510)        STATN(3),(SPCOV(3,I,NSTA),I=1,3),
     .                           IDMS1,(SACOV(1,I,NSTA),I=1,3),
     .                           IDMS2,(SACOV(2,I,NSTA),I=1,3),
     .                           IDMS3,(SACOV(3,I,NSTA),I=1,3)
 1140 LNCTR=LNCTR+4
C
C  LOCAL UNITS; CHECK TO PUNCH THE CAMERA STATION
```

```
C
 1150 IF (IPNST.NE.0) GO TO 1220
      DO 1160 I=1,3
           STATN(I+3)=PAKDMS(IDMSS(I))
 1160 CONTINUE
      WRITE (IP1,IOFM1) ID1,ID2,(STATN(I),I=1,3)
      WRITE (IP1,IOFM1) ID1,ID2,(STATN(I),I=4,6)
      GO TO 1220
C
C  GEOGRAPHIC UNITS; CHECK TO LIST THE CAMERA STATION
C
 1170 CALL RADDEG (STATN(1),IDMS4)
      CALL RADDEG (STATN(2),IDMS5)
      IF (ILTST.NE.0) GO TO 1200
      IF (IPROP.NE.0) GO TO 1180
      WRITE (IO,1520) IDMS4,IDMS1
      WRITE (IO,1530) ID1,ID2,IDMS5,IDMS2
      WRITE (IO,1540) STATN(3),IDMS3
      WRITE (IOS,2520) IDMS4,IDMS1
      WRITE (IOS,2530) ID1,ID2,IDMS5,IDMS2
      WRITE (IOS,2540) STATN(3),IDMS3
      GO TO 1190
 1180 IF (IEIGEN .NE. 0) GO TO 1188
C
C  Eigenvector/Eigenvalue Analysis & Output:
C
      DO 1181 I = 1, 3
      DO 1181 J = 1, 3
 1181 SPC(I, J) = SPCOV(I, J, NSTA)
      DO 1185 I = 1, 2
      DO 1185 J = 1, 3
      SPC(J, I) = SPC(J, I)*SPHRD(1)
 1185 SPC(I, J) = SPC(I, J)*SPHRD(1)
      CALL TRED2(3, 3, SPC, EVAL, EVX, EVEC)
      CALL TQL2 (3, 3, EVAL,  EVX, EVEC, IERR)
      DO 1186 I=1, 3
 1186 CALL RADDEG (DSQRT(SACOV(I,I,NSTA)), CH(I))
      WRITE(IO,1551)IDMS4,(EVEC(I,3),I=1,3),DSQRT(EVAL(3)),
     .              IDMS1,CH(1)
      WRITE(IO,1561)ID1,ID2,IDMS5,(EVEC(I,2),I=1,3),DSQRT(EVAL(2)),
     .              IDMS2,CH(2)
      WRITE(IO,1571)STATN(3),(EVEC(I,1),I=1,3),DSQRT(EVAL(1)),
     .              IDMS3,CH(3)
      WRITE(IOS,2551)IDMS4,(EVEC(I,3),I=1,3),DSQRT(EVAL(3))
      WRITE(IOS,2561)ID1,ID2,IDMS5,(EVEC(I,2),I=1,3),DSQRT(EVAL(2))
      WRITE(IOS,2571)STATN(3),(EVEC(I,1),I=1,3),DSQRT(EVAL(1)),
     .              IDMS1,CH(1), IDMS2,CH(2), IDMS3,CH(3)
      GO TO 1190
C
C  Covariance Output:
C
 1188 WRITE (IO,1550)         IDMS4,(SPCOV(1,I,NSTA),I=1,3),
     .                        IDMS1,(SACOV(1,I,NSTA),I=1,3)
      WRITE (IO,1560) ID1,ID2,IDMS5,(SPCOV(2,I,NSTA),I=1,3),
     .                        IDMS2,(SACOV(2,I,NSTA),I=1,3)
```

```
      WRITE (IO,1570)      STATN(3),(SPCOV(3,I,NSTA),I=1,3),
     .                        IDMS3,(SACOV(3,I,NSTA),I=1,3)
      WRITE (IOS,2550)        IDMS4,(SPCOV(1,I,NSTA),I=1,3)
      WRITE (IOS,2560)ID1,ID2,IDMS5,(SPCOV(2,I,NSTA),I=1,3)
      WRITE (IOS,2570)     STATN(3),(SPCOV(3,I,NSTA),I=1,3),
     .                        IDMS1,(SACOV(1,I,NSTA),I=1,3),
     .                        IDMS2,(SACOV(2,I,NSTA),I=1,3),
     .                        IDMS3,(SACOV(3,I,NSTA),I=1,3)
 1190 LNCTR=LNCTR+4
C
C  GEOGRAPHIC UNITS; CHECK TO PUNCH THE CAMERA STATION
C
 1200 IF (IPNST.NE.0) GO TO 1220
      STATN(1)=PAKDMS(IDMS4)
      STATN(2)=PAKDMS(IDMS5)
      DO 1210 I=1,3
           STATN(I+3)=PAKDMS(IDMSS(I))
 1210 CONTINUE
      WRITE (IP1,IOFM1) ID1,ID2,(STATN(I),I=1,3)
      WRITE (IP1,IOFM1) ID1,ID2,(STATN(I),I=4,6)
C
C  CHECK IF FINAL CAMERA STATION HAS BEEN PROCESSED
C
 1220 IF (NSTA.NE.NCAM) GO TO 1060
      IF (IPNST.NE.0) GO TO 1230
      WRITE (IP1,IOFM1) IEND,IEND
C
C
 1230 IF (ILTST.NE.0.OR.IPROP.EQ.0) GO TO 1250
      LNCTR=LNCTR+8
      IF (LNCTR.LT.MAXLIN) GO TO 1240
      CALL NEWPAG
      LNCTR=8
 1240 WRITE (IO,1580)
      WRITE (IOS,2580)
      CALL RADDEG (SSCVA(1),IDMS3)
      CALL RADDEG (SSCVA(2),IDMS4)
      CALL RADDEG (SSCVA(3),IDMS5)
      IF (IUNIT.EQ.0) THEN
C
C  WRITE CAM. STA. RMS OF: X, OMEGA, # PHOTOS, Y, PHI, Z, KAPPA
C
      WRITE (IO,1590) SSCVP(1),IDMS3,NCAM,SSCVP(2),IDMS4,SSCVP(3),IDMS5
      WRITE (IOS,2590)SSCVP(1),IDMS3,NCAM,SSCVP(2),IDMS4,SSCVP(3),IDMS5
      ELSE
           CALL RADDEG (SSCVP(1),IDMS1)
           CALL RADDEG (SSCVP(2),IDMS2)
C
C  WRITE CAM. STA. RMS OF: LNG, OMEGA, # PHOTOS, LAT, PHI, ELEV, KAPPA
C
           WRITE (IO,1600) IDMS1,IDMS3,NCAM,IDMS2,IDMS4,SSCVP(3),IDMS5
           WRITE (IOS,2600)IDMS1,IDMS3,NCAM,IDMS2,IDMS4,SSCVP(3),IDMS5
      END IF
C
C  BEGIN TO PROCESS THE OBJECT POINTS
```

```
C
C  INITIALIZATION FOR OBJECT POINTS
C
 1250 NSTA=0
      LNCTR=80
      DO 1260 I=1,3
           SSCVP(I)=ZERO
           NOSS(I)=0
 1260 CONTINUE
C
C  CHECK OPTION OF LISTING OBJECT POINTS
C
 1270 NSTA=NSTA+1
      IF (ILTGP.NE.0) GO TO 1290
C
C  CHECK TO LIST THE PAGE HEADING
C
      IF (LNCTR.LT.MAXLIN) GO TO 1300
      CALL NEWPAG
      WRITE (IO,1610)
      WRITE (IOS,2610)
      LNCTR=4
      IF (IPROP.NE.0) GO TO 1280
      IF (IUNIT.NE.0) GO TO 1275
      WRITE (IO,1620)
      WRITE (IOS,2620)
      GO TO 1300
 1275 WRITE (IO,1621)
      WRITE (IOS,2621)
      GO TO 1300
 1280 IF (IUNIT.NE.0) GO TO 1285
      IF (IEIGEN.NE.0) THEN
          WRITE (IO,1630)
          WRITE (IOS,2630)
      ENDIF
      IF (IEIGEN.EQ.0) THEN
          WRITE (IO,1632)
          WRITE (IOS,2632)
      ENDIF
      GO TO 1300
 1285 IF (IEIGEN.NE.0) THEN
          WRITE (IO,1631)
          WRITE (IOS,2631)
      ENDIF
      IF (IEIGEN.EQ.0) THEN
          WRITE (IO,1633)
          WRITE (IOS,2633)
      ENDIF
      GO TO 1300
C
C  CHECK OPTION OF PUNCHING OBJECT POINTS
C
 1290 IF (IPNGP.NE.0) GO TO 1420
C
C  READ A OBJECT POINT AND CHECK ITS UNITS
```

```
C
 1300 READ (ITAPE) ID1,ID2,IFLG,OBJECT,GPCOV
      IF(IFLG .LE. 6) NCNTRL=1
      IF (IPROP.EQ.0) GO TO 1320
      DO 1310 I=1,3
           DO 1310 J=1,3
           CONST=GPCOV(I,J)*SS
           GPCOV(I,J)=CONST
           IF (I.NE.J) GO TO 1310
           STATN(I)=DSQRT(CONST)
           IF (BTEST(IFLG,I-1)) THEN
                SSCVP(I)=SSCVP(I)+CONST
                NOSS(I)=NOSS(I)+1
           END IF
 1310 CONTINUE
 1320 IFLG=IFLG+1
      IF (IUNIT.NE.0) GO TO 1350
C
C  LOCAL UNITS; CHECK TO LIST THE OBJECT POINT
C
C  ANTHROPOMETRY OUTPUT
C
      IF(IANTH.NE.0)CALL STUFFP(ID1, ID2, OBJECT)
      IF (ILTGP.NE.0) GO TO 1340
      IF (IPROP.NE.0) GO TO 1330
      WRITE (IO,1640) ID1,ID2,INDTYP(IFLG),OBJECT
      WRITE (IOS,2640) ID1,ID2,INDTYP(IFLG),OBJECT
      LNCTR=LNCTR+1
      GO TO 1340
 1330 IF(IEIGEN.NE.0)GO TO 1338
      CALL TRED2(3, 3, GPCOV, EVAL, EVX, EVEC)
      CALL TQL2 (3, 3, EVAL, EVX, EVEC, IERR)
      WRITE (IO,1650) OBJECT(1),(EVEC(I,3),I=1,3),DSQRT(EVAL(3))
      WRITE (IO,1660) ID1,ID2,INDTYP(IFLG),OBJECT(2),(EVEC(I,2),I=1,3),
     .                DSQRT(EVAL(2))
      WRITE (IO,1670) OBJECT(3),(EVEC(I,1),I=1,3),DSQRT(EVAL(1))
      WRITE (IOS,2650) OBJECT(1),(EVEC(I,3),I=1,3),DSQRT(EVAL(3))
      WRITE (IOS,2660)ID1,ID2,INDTYP(IFLG),OBJECT(2),(EVEC(I,2),I=1,3),
     .                DSQRT(EVAL(2))
      WRITE (IOS,2670) OBJECT(3),(EVEC(I,1),I=1,3),DSQRT(EVAL(1))
      GO TO 1339
 1338 WRITE (IO,1650)                        OBJECT(1),(GPCOV(1,I),I=1,3),
     .                                        STATN(1)
      WRITE (IO,1660) ID1,ID2,INDTYP(IFLG),OBJECT(2),(GPCOV(2,I),I=1,3),
     .                                        STATN(2)
      WRITE (IO,1670)                        OBJECT(3),(GPCOV(3,I),I=1,3),
     .                                        STATN(3)
      WRITE (IOS,2650)                       OBJECT(1),(GPCOV(1,I),I=1,3),
     .                                        STATN(1)
      WRITE (IOS,2660)ID1,ID2,INDTYP(IFLG),OBJECT(2),(GPCOV(2,I),I=1,3),
     .                                        STATN(2)
      WRITE (IOS,2670)                       OBJECT(3),(GPCOV(3,I),I=1,3),
     .                                        STATN(3)
 1339 LNCTR=LNCTR+4
C
```

```
C  LOCAL UNITS; CHECK TO PUNCH THE OBJECT POINT
C
 1340 IF (IPNGP.NE.0) GO TO 1380
      WRITE (IP2,IOFM2) ID1,ID2,OBJECT
      GO TO 1380
C
C  GEOGRAPHIC UNITS; CHECK TO LIST THE OBJECT POINT
C
 1350 CALL RADDEG (OBJECT(1),IDMS1)
      CALL RADDEG (OBJECT(2),IDMS2)
      IF (ILTGP.NE.0) GO TO 1370
      IF (IPROP.NE.0) GO TO 1360
      WRITE (IO,1680) ID1,ID2,INDTYP(IFLG),IDMS1,IDMS2,OBJECT(3)
      WRITE (IOS,2680) ID1,ID2,INDTYP(IFLG),IDMS1,IDMS2,OBJECT(3)
      LNCTR=LNCTR+1
      GO TO 1370
 1360 IF(IEIGEN.NE.0)GO TO 1368
      DO 1365 I=1, 2
      DO 1365 J=1, 3
      GPCOV(J, I)=GPCOV(J, I)*SPHRD(1)
 1365 GPCOV(I, J)=GPCOV(I, J)*SPHRD(1)
      CALL TRED2(3, 3, GPCOV, EVAL, EVX, EVEC)
      CALL TQL2 (3, 3, EVAL, EVX, EVEC, IERR)
      WRITE (IO,1691) IDMS1,(EVEC(I,3),I=1,3),DSQRT(EVAL(3))
      WRITE (IO,1701) ID1,ID2,INDTYP(IFLG),IDMS2,(EVEC(I,2),I=1,3),
     .                DSQRT(EVAL(2))
      WRITE (IO,1711) OBJECT(3),(EVEC(I,1),I=1,3),DSQRT(EVAL(1))
      WRITE (IOS,2691)         IDMS1,(EVEC(I,3),I=1,3),DSQRT(EVAL(3))
      WRITE (IOS,2701) ID1,ID2,IDMS2,(EVEC(I,2),I=1,3),DSQRT(EVAL(2))
      WRITE (IOS,2711) INDTYP(IFLG),OBJECT(3),(EVEC(I,1),I=1,3),
     .                                                DSQRT(EVAL(1))
      GO TO 1369
 1368 CALL RADDEG (STATN(1),IDMS3)
      CALL RADDEG (STATN(2),IDMS4)
      WRITE (IO,1690)                     IDMS1,(GPCOV(1,I),I=1,3),
     .                                    IDMS3
      WRITE (IO,1700) ID1,ID2,INDTYP(IFLG),IDMS2,(GPCOV(2,I),I=1,3),
     .                                    IDMS4
      WRITE (IO,1710)                 OBJECT(3),(GPCOV(3,I),I=1,3),
     .                                 STATN(3)
      WRITE (IOS,2690)                    IDMS1,(GPCOV(1,I),I=1,3),
     .                                    IDMS3
      WRITE (IOS,2700) ID1,ID2,           IDMS2,(GPCOV(2,I),I=1,3),
     .                                    IDMS4
      WRITE (IOS,2710)   INDTYP(IFLG), OBJECT(3),(GPCOV(3,I),I=1,3),
     .                                 STATN(3)
 1369 LNCTR=LNCTR+4
C
C  GEOGRAPHIC UNITS; CHECK TO PUNCH THE OBJECT POINT
C
 1370 IF (IPNGP.NE.0) GO TO 1380
      OBJECT(1)=PAKDMS(IDMS1)
      OBJECT(2)=PAKDMS(IDMS2)
      WRITE (IP2,IOFM2) ID1,ID2,OBJECT
C
```

```
C  CHECK IF FINAL OBJECT POINT HAS BEEN PROCESSED
C
 1380 IF (NSTA.NE.NGPS) GO TO 1270
      IF (IPNGP.NE.0) GO TO 1390
      WRITE (IP2,IOFM2) IEND,IEND
C
C
 1390 IF (ILTGP.NE.0.OR.IPROP.EQ.0) GO TO 1420
      LNCTR=LNCTR+8
      IF (LNCTR.LT.MAXLIN) GO TO 1400
      CALL NEWPAG
      LNCTR=8
 1400 WRITE (IO,1720)
      WRITE (IOS,2720)
      DO 1410 I=1,3
           IF (NOSS(I).EQ.0) GO TO 1410
           SSCVP(I)=SQRT(SSCVP(I)/FLOAT(NOSS(I)))
 1410 CONTINUE
      IF (IUNIT.EQ.0) THEN
           WRITE (IO,1730) NOSS(1),SSCVP(1),NOSS(2),SSCVP(2),NOSS(3),
     .                                                     SSCVP(3)
           WRITE (IOS,2730)NOSS(1),SSCVP(1),NOSS(2),SSCVP(2),NOSS(3),
     .                                                     SSCVP(3)
      ELSE
           CALL RADDEG (SSCVP(1),IDMS1)
           CALL RADDEG (SSCVP(2),IDMS2)
           WRITE (IO,1740) NOSS(1),IDMS1,NOSS(2),IDMS2,NOSS(3),SSCVP(3)
           WRITE (IOS,2740)NOSS(1),IDMS1,NOSS(2),IDMS2,NOSS(3),SSCVP(3)
      END IF
 1420 RETURN
C
C  The following FORMAT Statements are for 132-column listings:
C
 1430 FORMAT (38X,'T R I A N G U L A T E D   C A M E R A   S T A T I O N
     . S'/)
 1440 FORMAT ('0',31X,'IDENT',11X,'POSITION',14X,'ATT',A17)
 1450 FORMAT ('0',3X,'IDENT',11X,'POSITION',14X,'COVARIANCE MATRIX',15X,
     .                           'ATT',A17,11X,'COVARIANCE MATRIX')
 1455 FORMAT ('0',3X,'IDENT',11X,'POSITION',10X,'ERROR ELLIPSOID',
     .' ORIENTATION    --->     LENGTH      ATT',A17,6X,'STD DEVIATION')
 1132 FORMAT ('0',15X,  'X =',F12.4,' m. ',SP,1P3D11.3,' ---> ',
     .       SS,0PF8.4,' m.','    OMEGA =',A15,3X,A15)
 1134 FORMAT (2X,2A4,6X,'Y =',F12.4,' m. ',SP,1P3D11.3,' ---> ',
     .       SS,0PF8.4,' m.','    PHI   =',A15,3X,A15)
 1136 FORMAT (16X,      'Z =',F12.4,' m. ',SP,1P3D11.3,' ---> ',
     .       SS,0PF8.4,' m.','    KAPPA =',A15,3X,A15)
 1460 FORMAT ('0',45X,'X =',F12.4,' m.',5X,'OMEGA =',A15)
 1470 FORMAT (29X,2A4,9X,'Y =',F12.4,' m.',5X,'PHI   =',A15)
 1480 FORMAT (46X,'Z =',F12.4,' m.',5X,'KAPPA =',A15)
 1490 FORMAT ('0',15X,'X =',F12.4,' m. ',SP,1P3D11.3,5X,'OMEGA =',
     .          A15,1X,3(1X,1PE10.3))
 1500 FORMAT (2X,2A4,6X,'Y =',F12.4,' m. ',SP,1P3D11.3,5X,'PHI   ='
     .         ,A15,1X,3(1X,1PE10.3))
 1510 FORMAT (16X,'Z =',F12.4,' m. ',SP,1P3D11.3,5X,'KAPPA =',
     .          A15,1X,3(1X,1PE10.3))
```

```
 1520 FORMAT ('0',40X,'LNG =',A15,8X,'OMEGA =',A15)
 1530 FORMAT (29X,2A4,4X,'LAT =',A15,8X,'PHI   =',A15)
 1540 FORMAT (41X,'ELV =',F15.4,' m.',5X,'KAPPA =',A15)
 1551 FORMAT ('0',12X,  'LNG =',A15,  1X,SP,1P3D11.3,' ---> ',
     .          SS, 0PF8.4, ' m.','   OMEGA =',A15,3X,A15)
 1561 FORMAT (1X,2A4,4X,'LAT =',A15,  1X,SP,1P3D11.3,' ---> ',
     .          SS, 0PF8.4, ' m.','   PHI   =',A15,3X,A15)
 1571 FORMAT (13X,        'ELV =',F15.4,1X,SP,1P3D11.3,' ---> ',
     .          SS, 0PF8.4, ' m.','   KAPPA =',A15,3X,A15)
 1550 FORMAT ('0',12X,'LNG =',A15,1X,SP,1P3D11.3,6X,'OMEGA =',
     .         A15,1X,3(1X,1PE10.3))
 1560 FORMAT (1X,2A4,4X,'LAT =',A15,1X,SP,1P3D11.3,6X,'PHI   =',
     .         A15,1X,3(1X,1PE10.3))
 1570 FORMAT (13X,'ELV =',F15.4,1X,SP,1P3D11.3,6X,'KAPPA =',
     .         A15,1X,3(1X,1PE10.3))
C
 1580 FORMAT (//25X,'S U M M A R Y   S T A T I S T I C S   F O R   C
     .A M E R A   S T A T I O N S'//65X,'RMS FOR STANDARD DEVIATIONS'/)
 1590 FORMAT (56X,'X =',F11.4,' m.',5X,'OMEGA =',A15,/40X,'COUNT =',I4,
     .          5X,'Y =',F11.4,' m.',5X,'PHI   =',A15,/56X,'Z =',F11.4,
     .          ' m.',5X,'KAPPA =',A15)
 1600 FORMAT (55X,'LNG =',A15,5X,'OMEGA =',A15,/39X,'COUNT =',I4,5X,'LAT
     . =',A15,5X,'PHI   =',A15,/55X,'ELV =',F15.4,' m.  KAPPA =',A15)
C
 1610 FORMAT(40X,'T R I A N G U L A T E D   O B J E C T   P O I N T S'/)
 1620 FORMAT (24X,'IDENT',33X,'POSITION (meters)'/)
 1621 FORMAT (24X,'IDENT',38X,'POSITION'/)
 1630 FORMAT (18X,'IDENT',9X,'POSITION (meters)',17X,'COVARIANCE MATRIX'
     .        ,14X,'STANDARD DEV (m)')
 1631 FORMAT (18X,'IDENT',14X,'POSITION',21X,'COVARIANCE MATRIX',16X,
     .                                                  'STANDARD DEV')
 1632 FORMAT (18X,'IDENT',9X,'POSITION (meters)',12X,
     .        'ERROR ELLIPSOID ORIENTATION    --->    LENGTH (m)')
 1633 FORMAT (18X,'IDENT',14X,'POSITION',13X,
     .        'ERROR ELLIPSOID ORIENTATION     --->   LENGTH (m)')
 1640 FORMAT (20X,2A4,2X,A3,12X,'X =',F12.4,4X,'Y =',F12.4,4X,'Z =',
     .                                                        F12.4)
 1650 FORMAT ('0',32X,'X =',F12.4,9X,SP,1P3D11.3,6X,S,0PF12.4)
 1660 FORMAT (15X,2A4,2X,A3,5X,'Y =',F12.4,9X,SP,1P3D11.3,
     .                                                  6X,S,0PF12.4)
 1670 FORMAT (     33X,'Z =',F12.4,9X,SP,1P3D11.3,6X,S,0PF12.4)
 1680 FORMAT (21X,2A4,2X,A3,3X,'LNG =',A15,7X,'LAT =',A15,7X,'ELV =',
     .                                                 F12.3,' (m.)')
 1690 FORMAT ('0',30X,'LNG =',A15,6X,SP,1P3D11.3,6X,A15)
 1700 FORMAT (15X,2A4,2X,A3,3X,'LAT =',A15,6X,SP,1P3D11.3,6X,A15)
 1710 FORMAT (31X,'ELV =',F15.4,' m    ',SP,1P3D11.3,6X,S,0PF15.4)
 1691 FORMAT ('0',30X,'LNG =',A15,6X,SP,1P3D11.3,S,0PF18.4)
 1701 FORMAT(15X,2A4,2X,A3,3X,'LAT =',A15,6X,SP,1P3D11.3,S,0PF18.4)
 1711 FORMAT (31X,'ELV =',F15.4,' m    ',SP,1P3D11.3,S,0PF18.4)
C
 1720 FORMAT (/27X,'S U M M A R Y   S T A T I S T I C S   F O R   O B
     . J E C T   P O I N T S'//49X,'RMS FOR STANDARD DEVIATIONS'/)
 1730 FORMAT (45X,'COUNT =',I4,5X,'X =',F15.4,' meters'/45X,'COUNT =',
     .                    I4,5X,'Y =',F15.4,' meters'/45X,'COUNT =',
     .                                  I4,5X,'Z =',F15.4,' meters')
```

```
 1740 FORMAT(45X,'COUNT =',I4,5X,'LNG =',A15/45X,'COUNT =',I4,5X,'LAT ='
     .                  ,A15/45X,'COUNT =',I4,5X,'ELV =',F15.4,' meters')
C
C  The following FORMAT Statements are for 80-column listings:
C
 2430 FORMAT (10X,'T R I A N G U L A T E D   C A M E R A   S T A T I O N
     . S'/31X,A17)
 2440 FORMAT ('0',10X,'Ident',11X,'Position',17X,'Attitude')
 2450 FORMAT ('0   Ident',7X,'Position/Attitude',9X,'Covariance Matrix')
 2455 FORMAT ('0   Ident',11X,'Position',12X,'Error Ellipsoid',
     .'      --->    Length')
 2132 FORMAT('0',15X,'X =',F12.4,' m. ',SP,3F8.4,' ---> ',SS,F8.4,' m.')
 2134 FORMAT (2X,2A4,6X,'Y =',F12.4,' m. ',SP,3F8.4,' ---> ',SS,F8.4,
     .          ' m.')
 2136 FORMAT (16X,'Z =',F12.4,' m. ',SP,3F8.4,' ---> ',SS,F8.4,' m.'//
     .16X,'             Omega =',A15,'           ',A15/
     .16X,'Attitude:  Phi   =',A15,'  Std Dev:',A15/
     .16X,'             Kappa =',A15,'           ',A15)
 2460 FORMAT ('0',22X,'X =',F12.4,' m.',5X,'Omega =',A15)
 2470 FORMAT (6X,2A4,9X,'Y =',F12.4,' m.',5X,'Phi   =',A15)
 2480 FORMAT (23X,'Z =',F12.4,' m.',5X,'Kappa =',A15)
 2490 FORMAT ('0',15X,  'X =',F12.4,' m. ',SP,1P3D11.3)
 2500 FORMAT (2X,2A4,6X,'Y =',F12.4,' m. ',SP,1P3D11.3)
 2510 FORMAT (16X,      'Z =',F12.4,' m. ',SP,1P3D11.3/
     .              12X,'Omega =',A15,1X,1P3D11.3/
     .              12X,'Phi   =' A15,1X,1P3D11.3/
     .              12X,'Kappa =',A15,1X,1P3D11.3)
 2520 FORMAT ('0',19X,  'Lng =',A15,        8X,'Omega =',A15)
 2530 FORMAT (8X,2A4,4X,'Lat =',A15,        8X,'Phi   =',A15)
 2540 FORMAT (20X,      'Elv =',F15.4,' m.',5X,'Kappa =',A15)
 2551 FORMAT ('0',12X,  'Lng =',A15,  1X,SP,3F8.4,'  ---> ',SS,F9.4,'m')
 2561 FORMAT (1X,2A4,4X,'Lat =',A15,  1X,SP,3F8.4,'  ---> ',SS,F9.4,'m')
 2571 FORMAT (13X,      'Elv =',F15.4,1X,SP,3F8.4,'  ---> ',SS,F9.4,'m'
     .               /11X,'Omega =',A15,11X,A15/
     .                11X,'Phi   =',A15,'  Std. Dev.',A15/
     .                11X,'Kappa =',A15,11X,A15)
 2550 FORMAT ('0',12X,  'Lng =',A15,  1X,SP,1P3D11.3)
 2560 FORMAT (1X,2A4,4X,'Lat =',A15,  1X,SP,1P3D11.3)
 2570 FORMAT (13X,      'Elv =',F15.4,1X,SP,1P3D11.3/
     .            11X,    'Omega =',A15,  1X,1P3D11.3/
     .            11X,    'Phi   =',A15,  1X,1P3D11.3/
     .            11X,    'Kappa =',A15,  1X,1P3D11.3)
C
 2580 FORMAT (//' S U M M A R Y   S T A T I S T I C S   F O R   C A',
     .' M E R A   S T A T I O N S'//26X,'RMS For Standard Deviations'/)
 2590 FORMAT (27X,'X =',F11.4,' m.',5X,'Omega =',A15,/11X,'Count =',I4,
     .            5X,'Y =',F11.4,' m.',5X,'Phi   =',A15,/27X,'Z =',F11.4,
     .            ' m.',5X,'Kappa =',A15)
 2600 FORMAT (21X,'Lng =',A15,5X,'Omega =',A15,/5X,'Count =',I4,5X,'Lat
     .=',A15,5X,'Phi   =',A15,/21X,'Elv =',F15.4,' m.  Kappa =',A15)
C
 2610 FORMAT(14X,'T R I A N G U L A T E D   O B J E C T   P O I N T S'/)
 2620 FORMAT (7X,'Ident',25X,'Position (meters)'/)
 2621 FORMAT (3X,'Ident',38X,'Position'/)
 2630 FORMAT ('  Ident',7X,'Position (meters)',9X,'Covariance Matrix'
```

```
     .         ,9X,'Std Dev (m)')
 2631 FORMAT ('  Ident',12X,'Position',13X,'Covariance Matrix',10X,
     .                                                     'Std Dev')
 2632 FORMAT ('  Ident',7X,'Position (meters)',11X,
     .         'Error Ellipsoid --->    Length (m)')
 2633 FORMAT (' Ident',14X,'Position',11X,
     .         'Error Ellipsoid --->     Length (m)')
 2640 FORMAT (7X,2A4,2X,A3,'  X =',F12.4,3X,'Y =',F12.4,3X,'Z =',F12.4)
 2650 FORMAT ('0',      13X,' X =',F12.4,2X,SP,1P3D11.3,2X,SS,0PF8.4)
 2660 FORMAT (1X,2A4,2X,A3,' Y =',F12.4,2X,SP,1P3D11.3,2X,SS,0PF8.4)
 2670 FORMAT (           14X,' Z =',F12.4,2X,SP,1P3D11.3,2X,SS,0PF8.4)
 2680 FORMAT (2X,2A4,X,A3,X,'Lng =',A15,2X,'Lat =',A15,2X,'Elv =',
     .                                                      F8.4,' m.')
 2690 FORMAT ('0            Lng =',A15,     X,SP,1P3D11.3,  A15)
 2700 FORMAT (1X,2A4,     ' Lat =',A15,     X,SP,1P3D11.3,  A15)
 2710 FORMAT (6X, A3,     ' Elv =',F15.4,'m',SP,1P3D11.3,S,0PF15.4)

 2691 FORMAT ('0               Lng = ',A15,      3X,SP,3F8.4,S,F12.4)
 2701 FORMAT (1X,2A4,    '     Lat = ',A15,      3X,SP,3F8.4,S,F12.4)
 2711 FORMAT (6X, A3,    '     Elv = ',F15.4,'m  ',SP,3F8.4,S,F12.4)
C
 2720 FORMAT (/'     S U M M A R Y    S T A T I S T I C S    F O R    O B'
     . ' J E C T    P O I N T S'//28X,'RMS For Standard Deviations'/)
 2730 FORMAT (21X,'Count =',I4,5X,'X =',F15.4,' meters'/21X,'Count =',
     .                      I4,5X,'Y =',F15.4,' meters'/21X,'Count =',
     .                                      I4,5X,'Z =',F15.4,' meters')
 2740 FORMAT(21X,'Count =',I4,5X,'Lng =',A15/21X,'Count =',I4,5X,'Lat ='
     .                  ,A15/21X,'Count =',I4,5X,'Elv =',F15.4,' meters')
      END


      SUBROUTINE LSTGRS (ITAPE)
C
C   THIS SUBROUTINE LISTS OBJECT CONTROL RESIDUALS.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*15 IDMS
      COMMON /TAPES/  IN,IO,IOS,IDUM(14)
      INCLUDE 'PARAMS.INC'
      INCLUDE 'OPTION.INC'
      DIMENSION       XYZ(3,4), IXYZ(6,4), MARKS(2,3,4), IDS(2,4)
      DIMENSION       SSQ(3), IRMS(3), ITAB(9,ISZ3)
      DIMENSION       KEYS(2), IDMS(1,2,4), NAME1(3), NAME2(3)
      EQUIVALENCE     (XYZ(1,1),IXYZ(1,1))
      DATA NKEY,KEYS    /2,1,2/
      DATA NAME1        /' X =',' Y =',' Z ='/
      DATA NAME2        /'LNG=','LAT=','ELV='/
      DATA IBLANK,ML,MR /' ','(',')'/
      DATA ZERO,MAXLIN  /0.0D0,56/
C
C   READ AND SORT OBJECT RESIDUALS.
C
      REWIND ITAPE
      NP=1
 1010 READ (ITAPE,END=1020) (ITAB(I,NP),I=1,9)
```

```
      NP=NP+1
      GO TO 1010
 1020 NP=NP-1
      IF (NP.EQ.0) RETURN
      CALL SORTER (ITAB,9,NP,KEYS,NKEY)
      LINE=100
      DO 1030 I=1,3
           SSQ(I)=ZERO
           IRMS(I)=0
 1030 CONTINUE
      N=0
      DO 1110 II=1,NP
           N=N+1
           IDS(1,N)=ITAB(1,II)
           IDS(2,N)=ITAB(2,II)
           IND=ITAB(3,II)
           DO 1040 I=1,6
                IXYZ(I,N)=ITAB(I+3,II)
 1040      CONTINUE
           DO 1060 I=1,3
                J=MOD(IND,2)
                IND=(IND-J)/2
                IF (J.EQ.0) GO TO 1050
                MARKS(1,I,N)=ML
                MARKS(2,I,N)=MR
                GO TO 1060
 1050           MARKS(1,I,N)=IBLANK
                MARKS(2,I,N)=IBLANK
                SSQ(I)=SSQ(I)+XYZ(I,N)**2
                IRMS(I)=IRMS(I)+1
 1060      CONTINUE
           IF (N.NE.4.AND.N.NE.2.AND.II.NE.NP) GO TO 1110
           N1=N-1
           IF(N.EQ.1)N1=1
C
C  PRINT CONTENTS OF BUFFER.
C
           IF (LINE.LT.MAXLIN) GO TO 1070
           LINE=4
           CALL NEWPAG
           WRITE (IO,1140)
           WRITE (IOS,2140)
 1070      LINE=LINE+4
           IF (IUNIT.EQ.0) GO TO 1090
           DO 1080 I=1,N
                DO 1080 J=1,2
                CALL RADDEG (XYZ(J,I),IDMS(1,J,I))
 1080      CONTINUE
           IF (N.EQ.4.OR.II.EQ.NP) THEN
           WRITE (IO,1150)                 (NAME2(1),MARKS(1,1,J),
     .                                   IDMS(1,1,J),MARKS(2,1,J),J=1,N)
           WRITE (IO,1160)((IDS(I,J),I=1,2),NAME2(2),MARKS(1,2,J),
     .                                   IDMS(1,2,J),MARKS(2,2,J),J=1,N)
           WRITE (IO,1170)                 (NAME2(3),MARKS(1,3,J),
     .                                     XYZ(3,J),MARKS(2,3,J),J=1,N)
```

```
          ENDIF
          WRITE (IOS,2150)                         (NAME2(1),MARKS(1,1,J),
     .                                      IDMS(1,1,J),MARKS(2,1,J),J=N1,N)
          WRITE (IOS,2160)((IDS(I,J),I=1,2),NAME2(2),MARKS(1,2,J),
     .                                      IDMS(1,2,J),MARKS(2,2,J),J=N1,N)
          WRITE (IOS,2170)                         (NAME2(3),MARKS(1,3,J),
     .                                         XYZ(3,J),MARKS(2,3,J),J=N1,N)
          GO TO 1100
 1090     CONTINUE
          IF (N.EQ.4.OR.II.EQ.NP) THEN
          WRITE (IO,1180)                         (NAME1(1),MARKS(1,1,J),
     .                                          XYZ(1,J),MARKS(2,1,J),J=1,N)
          WRITE (IO,1190)((IDS(I,J),I=1,2),NAME1(2),MARKS(1,2,J),
     .                                          XYZ(2,J),MARKS(2,2,J),J=1,N)
          WRITE (IO,1180)                         (NAME1(3),MARKS(1,3,J),
     .                                          XYZ(3,J),MARKS(2,3,J),J=1,N)
          ENDIF
          WRITE (IOS,2180)                         (NAME1(1),MARKS(1,1,J),
     .                                         XYZ(1,J),MARKS(2,1,J),J=N1,N)
          WRITE (IOS,2190)((IDS(I,J),I=1,2),NAME1(2),MARKS(1,2,J),
     .                                         XYZ(2,J),MARKS(2,2,J),J=N1,N)
          WRITE (IOS,2180)                         (NAME1(3),MARKS(1,3,J),
     .                                         XYZ(3,J),MARKS(2,3,J),J=N1,N)
 1100     N=4-N
          WRITE (IO,*)
          WRITE (IOS,*)
 1110 CONTINUE
C
C
      DO 1120 I=1,3
          SSQ(I)=DSQRT(SSQ(I)/DFLOAT(IRMS(I)))
 1120 CONTINUE
      IF (IUNIT.EQ.0) GO TO 1130
      CALL RADDEG (SSQ(1),IDMS(1,1,1))
      CALL RADDEG (SSQ(2),IDMS(1,2,1))
      WRITE (IO,1210)  (IRMS(I),IDMS(1,I,1),I=1,2),IRMS(3),SSQ(3)
      WRITE (IOS,2210)(IRMS(I),IDMS(1,I,1),I=1,2),IRMS(3),SSQ(3)
      RETURN
 1130 WRITE (IO,1220)  (IRMS(I),SSQ(I),I=1,3)
      WRITE (IOS,2220)(IRMS(I),SSQ(I),I=1,3)
C
      RETURN
C
C  The following FORMAT Statements are for 132-column listings:
C
 1140 FORMAT (28X,'C O R R E C T I O N S   A P P L I E D   T O   O B
     .J E C T   C O N T R O L'/)
 1150 FORMAT (4(12X,A4,A1,A15,A1))
 1160 FORMAT (4(2X,2A4,2X,A4,A1,A15,A1))
 1170 FORMAT (4(12X,A4,A1,F14.3,'m',A1))
 1180 FORMAT (1X,4(12X,A4,A1,F12.4,' m',A1))
 1190 FORMAT (1X,4(3X,2A4,1X,A4,A1,F12.4,' m',A1))
 1210 FORMAT (/35X,'LNG .... NUMBER OF COMPONENTS =',I5,4X,'RMS = ',A15,
     .    /35X,'LAT .... NUMBER OF COMPONENTS =',I5,4X,'RMS = ',A15,/35X,
     . 'ELV .... NUMBER OF COMPONENTS =',I5,4X,'RMS = ',F15.4,' meters')
```

```
 1220 FORMAT (/37X,'X .... NUMBER OF COMPONENTS =',I5,4X,'RMS = ',F14.4,
     .' meters'/37X,'Y .... NUMBER OF COMPONENTS =',I5,4X,'RMS = ',F14.4
     .,' meters'/37X,'Z .... NUMBER OF COMPONENTS =',I5,4X,'RMS = ',
     .F14.4,' meters')
C
C  The following FORMAT Statements are for 80-column listings:
C
 2140 FORMAT (2X,'C O R R E C T I O N S   A P P L I E D   T O   O B J
     . E C T   C O N T R O L'/)
 2150 FORMAT (9X,2(12X,       A4, A1,A15,       A1))
 2160 FORMAT (9X,2(2X,2A4,2X, A4, A1,A15,       A1))
 2170 FORMAT (9X,2(12X,       A4, A1,F14.3,'m', A1))
 2180 FORMAT (9X,2(12X,       A4, A1,F12.4,' m',A1))
 2190 FORMAT (9X,2(3X,2A4,1X, A4, A1,F12.4,' m',A1))
 2210 FORMAT (/9X,'Lng .... Number of Components =',I5,4X,'RMS = ',A15,
     .    /9X,'Lat .... Number of Components =',I5,4X,'RMS = ',A15,/9X,
     . 'Elv .... Number of Components =',I5,4X,'RMS = ',F15.4,' meters')
 2220 FORMAT (/10X,'X .... Number of Components =',I5,4X,'RMS = ',F14.4,
     .' meters'/10X,'Y .... Number of Components =',I5,4X,'RMS = ',F14.4
     .,' meters'/10X,'Z .... Number of Components =',I5,4X,'RMS = ',
     .F14.4,' meters')
      END


      SUBROUTINE SORTER (IARRAY,IROW,NARRAY,KEYS,NKEY)
C
C  THIS SUBROUTINE PERFORMS A GENERAL SORT OF A CORE-STORED
C              TWO-DIMENSIONAL, INTEGER ARRAY
C
      DIMENSION IARRAY(IROW,1), KEYS(1)
C
C  IARRAY = A TWO DIMENSIONAL ARRAY (IROW,---)
C  IROW   = DIMENSION OF FIRST SUBSCRIPT OF ARRAY IARRAY
C  NARRAY = NUMBER OF COLUMNS IN IARRAY
C  KEYS   = VECTOR OF INDICES FOR THE ROWS ON WHICH TO SORT
C  NKEY   = NUMBER OF ENTRIES IN VECTOR KEYS
C
C  CHECK SIZE OF ARRAY
C
      IF (NARRAY.LE.1) GO TO 1060
C
C  THIS LOOP PERFORMS A SORT ON EACH KEY ROW
C
      II=NKEY
 1010 IF (II.EQ.0) GO TO 1060
      KEY=KEYS(II)
      II=II-1
      IF (KEY.LT.0.OR.KEY.GT.IROW) THEN
          CALL CLR
          CALL BEEP
          CALL CURDWN (8)
          WRITE (*, 3000) KEY
          STOP
      ENDIF
C
```

```
C   THIS LOOP MOVES THE LARGEST ELEMENT TO THE BOTTOM OF THE ARRAY
C
       INDEX=NARRAY
C
C   PERFORM A MAXIMUM OF (NARRAY - 1) SORT PASSES
C
       DO 1050 JJ=2,NARRAY
            IF (INDEX.LE.1) GO TO 1010
            LAST=INDEX
            INDEX=0
C
C   THIS LOOP MOVES THE LARGEST ELEMENT TO THE BOTTOM OF THE ARRAY
C
            NUMOLD=IARRAY(KEY,1)
            DO 1040 KK=2,LAST
                 NUMNEW=IARRAY(KEY,KK)
                 IF (NUMOLD.LE.NUMNEW) GO TO 1030
                 INDEX=KK-1
C
C   EXCHANGE TWO COLUMNS
C
                 DO 1020 LL=1,IROW
                      NUMNEW=IARRAY(LL,INDEX)
                      IARRAY(LL,INDEX)=IARRAY(LL,KK)
                      IARRAY(LL,KK)=NUMNEW
 1020            CONTINUE
                 GO TO 1040
 1030            NUMOLD=NUMNEW
 1040       CONTINUE
 1050  CONTINUE
       GO TO 1010
 1060  RETURN
 3000  FORMAT (' ', 'SUBROUTINE SORTER FATAL ERROR: KEY = ', I4)
       END

       SUBROUTINE TRED2 (NM,N,A,D,E,Z)
C
C   This subroutine reduces a real symmetric matrix to a
C   symmetric tridiagonal matrix using and accumulating
C   orthogonal similarity transformations.  This reduced form and
C   the transformation matrix are used by SUBROUTINE TQL2 to find
C   the eigenvalues and eigenvectors of the original matrix.
C
C   On Input
C
C         NM must be set to the row dimension of two-dimensional
C            array parameters as declared in the calling program
C            dimension statement for A and Z.
C
C         N  is the order of the matrix, and must not be greater
C            than NM.
C
C         A  contains the real symmetric input matrix with row
C            dimension at least N to be reduced to tridiagonal form.
C            Only the full lower triangle of the matrix need be
```

```
C          supplied.
C
C  On Output
C
C        D  contains the diagonal elements of the tridiagonal
C           matrix of dimension of at least order N.
C
C        E  contains the subdiagonal elements of the tridiagonal
C           matrix in its last N-1 positions.  E(1) is set to zero.
C
C        Z  contains the orthogonal transformation matrix
C           produced in the reduction with row dimension NM and
C           column dimension at least N to the tridiagonal form.
C
C  A and Z may coincide.  if distinct, A is unaltered.
C
C  DISCUSSION OF METHOD AND ALGORITHM.
C
C  The lower triangle of A is initially copied into Z and all
C  subsequent operations are preformed on Z.
C
C  The tridiagonal reduction is performed in the following way.
C  Starting with J = N,  the elements in the J-th row to the
C  left of the diagonal are first scaled,  to avoid possible
C  underflow in the transformation that might result in severe
C  departure from orthognality.  The sum of squares SIGMA of
C  these scaled elements is next formed.  Then,  a vector U and
C  a scalar
C                                T
C                           H = U U/2
C
C  define an operator
C                                   T
C                         P = I - UU /H
C
C  which is orthogonal and symmetric and for which the
C  similarity transformation PAP eliminates the elements in
C  the J-th row of A to the left of the subdiagonal and the
C  symmetrical elements in the J-th column.
C
C  The non-zero components of U are the elements of the J-th
C  row to the left of the diagonal with the last of them
C  augmented by the square root of SIGMA prefixed by the sign
C  of the subdiagonal element.  By storing the transformed
C  subdiagonal element in E(J) and not overwriting the row
C  elements eliminated in the transformation,  full information
C  about P is saved for later accumulation of transformations.
C
C  The transformation sets E(J) equal to the square root of
C  SIGMA prefixed by sign opposite to that of the replaced
C  subdiagonal element.
C
C  The above steps are repeated on further rows of the
C  transformaed A in reverse order until A is reduced to
C  tridiagonal form; that is, repeated for J = N-1, N-2, ..., 3.
```

```
C
C  Finally, the orthogonal transformation matrix is accumulated
C  in Z as the product of the N-2 operators defined in the
C  tridiagonal reduction.
C
C
C  This subroutine is a translation of the ALGOL procedure TRED2,
C  NUM. MATH. 11, 181-195(1968) by Martin, Reinsch, and Wilkinson.
C  Handbook for Auto. Comp., Vol.II-LINEAR ALGEBRA, 212-226(1971).
C
      INTEGER          I,J,K,L,N,II,NM,JP1
      DOUBLE PRECISION A(NM,N),D(N),E(N),Z(NM,N)
      DOUBLE PRECISION F,G,H,HH,SCALE
      DO 1020 I=1,N
           DO 1010 J=I,N
 1010           Z(J,I)=A(J,I)
           D(I)=A(N,I)
 1020 CONTINUE
      IF (N.EQ.1) GO TO 1240
C
C  For I = N step -1 until 2 DO --
C
      DO 1170 II=2,N
           I=N+2-II
           L=I-1
           H=0.0D0
           SCALE=0.0D0
           IF (L.LT.2) GO TO 1040
C
C  Scale row (ALGOL TOL then not needed)
C
           DO 1030 K=1,L
 1030           SCALE=SCALE+DABS(D(K))
           IF (SCALE.NE.0.0D0) GO TO 1060
 1040      E(I)=D(L)
           DO 1050 J=1,L
                D(J)=Z(L,J)
                Z(I,J)=0.0D0
                Z(J,I)=0.0D0
 1050      CONTINUE
           GO TO 1160
 1060      DO 1070 K=1,L
                D(K)=D(K)/SCALE
                H=H+D(K)*D(K)
 1070      CONTINUE
           F=D(L)
           G=-DSIGN(DSQRT(H),F)
           E(I)=SCALE*G
           H=H-F*G
           D(L)=F-G
C
C  Form  A * U
C
           DO 1080 J=1,L
 1080           E(J)=0.0D0
```

```
          DO 1110 J=1,L
               F=D(J)
               Z(J,I)=F
               G=E(J)+Z(J,J)*F
               JP1=J+1
               IF (L.LT.JP1) GO TO 1100
               DO 1090 K=JP1,L
                    G=G+Z(K,J)*D(K)
                    E(K)=E(K)+Z(K,J)*F
 1090          CONTINUE
 1100          E(J)=G
 1110     CONTINUE
C
C  Form  P
C
          F=0.0D0
          DO 1120 J=1,L
               E(J)=E(J)/H
               F=F+E(J)*D(J)
 1120     CONTINUE
          HH=F/(H+H)
C
C  Form  Q
C
          DO 1130 J=1,L
 1130          E(J)=E(J)-HH*D(J)
C
C  Form reduced  A
C
          DO 1150 J=1,L
               F=D(J)
               G=E(J)
               DO 1140 K=J,L
 1140               Z(K,J)=Z(K,J)-F*E(K)-G*D(K)
               D(J)=Z(L,J)
               Z(I,J)=0.0D0
 1150     CONTINUE
 1160     D(I)=H
 1170 CONTINUE
C
C  Accumulation of transformation matrices
C
     DO 1230 I=2,N
          L=I-1
          Z(N,L)=Z(L,L)
          Z(L,L)=1.0D0
          H=D(I)
          IF (H.EQ.0.0D0) GO TO 1210
          DO 1180 K=1,L
 1180          D(K)=Z(K,I)/H
          DO 1200 J=1,L
               G=0.0D0
               DO 1190 K=1,L
 1190               G=G+Z(K,I)*Z(K,J)
               DO 1200 K=1,L
```

```
                Z(K,J)=Z(K,J)-G*D(K)
 1200       CONTINUE
 1210       DO 1220 K=1,L
 1220           Z(K,I)=0.0D0
 1230 CONTINUE
 1240 DO 1250 I=1,N
          D(I)=Z(N,I)
          Z(N,I)=0.0D0
 1250 CONTINUE
      Z(N,N)=1.0D0
      E(1)=0.0D0
      RETURN
      END


      SUBROUTINE TQL2 (NM,N,D,E,Z,IERR)
C
C  This subroutine finds the eigenvalues and eigenvectors
C  of a symmetric tridiagonal matrix by the QL method.
C  The eigenvectors of a full symmetric matrix can also
C  be found if  TRED2  has been used to reduce this
C  full matrix to tridiagonal form.
C
C  On Input
C
C         NM must be set to the row dimension of two-dimensional
C            array Z as specified in the DIMENSION statement for
C            Z in the calling program.
C
C          N is the order of the matrix, and must not be greater
C            than NM.
C
C          D contains the diagonal elements of the input
C            symmetric tridiagonal matrix.
C
C          E contains the subdiagonal elements of the input matrix
C            in its last N-1 positions.  E(1) is arbitrary.
C
C          Z is a two-dimensional variable with row dimension NM
C            and column dimension at least N.  If the eigenvectors
C            of the symmetric tridiagonal matrix are
C            desired,  then on input,  Z contains the
C            identity matrix of order N, on on output,  contains the
C            transformation matrix produced in TRED2 which reduced
C            the full matrix to tridiagonal form.
C
C  On Output
C
C          D contains the eigenvalues in ascending order.  if an
C            error exit is made, the eigenvalues are correct but
C            unordered for indices 1,2,...,IERR-1.
C
C          E has been destroyed.
C
C          Z contains orthonormal eigenvectors of the symmetric
```

```
C          tridiagonal (or full) matrix.  if an error exit is made,
C          Z contains the eigenvectors associated with the stored
C          eigenvalues.
C
C        IERR is set to
C          ZERO       for normal return,
C          J          if the J-th eigenvalue has not been
C                     determined after 30 iterations.
C
C              [Call PYTHAG for  DSQRT(A*A + B*B)]
C
C  DISCUSSION OF METHOD AND ALGORITHM.
C
C  The eigenvalues are determined by the QL method.  The
C  essence of this method is a process whereby a sequence of
C  symmetric tridiagonal matrices,  unitarily similar to the
C  original symmetric tridiagonal matrix,  is formed which
C  converges to a diagonal matrix.  The rate of convergence of
C  this sequence is improved by shifting the origin at each
C  iteration.  Before the iterations for each eigenvalue,  the
C  symmetric tridiagonal matrix is checked for a possible
C  splitting into submatrices.  If a splitting occurs,  only the
C  uppermost submatrix participates in the next iteration.  The
C  similarity transformations used in each iteration are
C  accumulated in the Z array, producing the orthonormal
C  eigenvectors for the original matrix.  Finally, the
C  eigenvalues are ordered in ascending order and the
C  eigenvectors are ordered consistently.
C
C  The origin shift at each iteration is the eigenvalue of the
C  current uppermost 2 X 2 principal minor closer to the first
C  diagonal element of this minor.  Whenever the uppermost 1 X 1
C  principal submatrix finally splits from the rest of the
C  matrix,  its element is taken to be an eigenvalue of the
C  original matrix and the algorithm proceeds with the remaining
C  submatrix.  This process is continued until the matrix has
C  split completely into submatrices of order 1.  The
C  tolerances in the splitting tests are proportional to the
C  relative machine precision.
C
C
C  This subroutine is a translation of the ALGOL procedure TQL2,
C  Num. Math. 11, 293-306(1968) by Bowdler, Martin, Reinsch, and
C  Wilkinson.
C  HANDBOOK FOR AUTO. COMP., VOL.II-Linear Algebra, 227-240(1971).
C
      INTEGER          I,J,K,L,M,N,II,L1,L2,NM,MML,IERR
      DOUBLE PRECISION D(N),E(N),Z(NM,N)
      DOUBLE PRECISION C,C2,C3,DL1,EL1,F,G,H,P,R,S,S2,
     .                 TST1,TST2,PYTHAG
      IERR=0
      IF (N.EQ.1) GO TO 1150
      DO 1010 I=2,N
 1010    E(I-1)=E(I)
      F=0.0D0
```

```
      TST1=0.0D0
      E(N)=0.0D0
      DO 1100 L=1,N
           J=0
           H=DABS(D(L))+DABS(E(L))
           IF (TST1.LT.H) TST1=H
C
C  Look for small sub-diagonal element
C
           DO 1020 M=L,N
                TST2=TST1+DABS(E(M))
                IF (TST2.EQ.TST1) GO TO 1030
C
C  E(N) is always zero, so there is no exit
C  through the bottom of the loop
C
 1020      CONTINUE
 1030      IF (M.EQ.L) GO TO 1090
 1040      IF (J.EQ.30) GO TO 1140
           J=J+1
C
C  Form shift
C
           L1=L+1
           L2=L1+1
           G=D(L)
           P=(D(L1)-G)/(2.0D0*E(L))
           R=PYTHAG(P,1.0D0)
           D(L)=E(L)/(P+DSIGN(R,P))
           D(L1)=E(L)*(P+DSIGN(R,P))
           DL1=D(L1)
           H=G-D(L)
           IF (L2.GT.N) GO TO 1060
           DO 1050 I=L2,N
 1050           D(I)=D(I)-H
 1060      F=F+H
C
C  QL transformation
C
           P=D(M)
           C=1.0D0
           C2=C
           EL1=E(L1)
           S=0.0D0
           MML=M-L
C
C  For I = M - 1 step -1 until 1 DO
C
           DO 1080 II=1,MML
                C3=C2
                C2=C
                S2=S
                I=M-II
                G=C*E(I)
                H=C*P
```

```
                    R=PYTHAG(P,E(I))
                    E(I+1)=S*R
                    S=E(I)/R
                    C=P/R
                    P=C*D(I)-S*G
                    D(I+1)=H+S*(C*G+S*D(I))
C
C   Form Vector
C
                    DO 1070 K=1,N
                         H=Z(K,I+1)
                         Z(K,I+1)=S*Z(K,I)+C*H
                         Z(K,I)=C*Z(K,I)-S*H
 1070               CONTINUE
 1080          CONTINUE
               P=-S*S2*C3*EL1*E(L)/DL1
               E(L)=S*P
               D(L)=C*P
               TST2=TST1+DABS(E(L))
               IF (TST2.GT.TST1) GO TO 1040
 1090          D(L)=D(L)+F
 1100 CONTINUE
C
C   Order Eigenvalues and Eigenvectors
C
      DO 1130 II=2,N
               I=II-1
               K=I
               P=D(I)
               DO 1110 J=II,N
                    IF (D(J).GE.P) GO TO 1110
                    K=J
                    P=D(J)
 1110          CONTINUE
               IF (K.EQ.I) GO TO 1130
               D(K)=D(I)
               D(I)=P
               DO 1120 J=1,N
                    P=Z(J,I)
                    Z(J,I)=Z(J,K)
                    Z(J,K)=P
 1120          CONTINUE
 1130 CONTINUE
      GO TO 1150
C
C   Set Error -- No convergence to an
C    Eigenvalue after 30 iterations
C
 1140 IERR=L
 1150 RETURN
      END


      DOUBLE PRECISION FUNCTION PYTHAG (A,B)
C
```

```
C  Finds DSQRT(A**2+B**2) without Overflow or destructive Underflow
C
      DOUBLE PRECISION A,B,P,R,S,T,U
      P=DMAX1(DABS(A),DABS(B))
      IF (P.EQ.0.0D0) GO TO 1020
      R=(DMIN1(DABS(A),DABS(B))/P)**2
 1010 T=4.0D0+R
      IF (T.EQ.4.0D0) GO TO 1020
      S=R/T
      U=1.0D0+2.0D0*S
      P=U*P
      R=(S/U)**2*R
      GO TO 1010
 1020 PYTHAG=P
      RETURN
      END
```

# PC Giant

## Source Code

## File Name: 4.FOR (Utilities)

## 14 June 1990

```
      SUBROUTINE MPYAB (A,B,C,L,M,N)
C
C  THIS PROGRAM PERFORMS THE FOLLOWING MATRIX OPERATION:
C                C(L,N) = A(L,M) * B(M,N)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1), B(1), C(1)
C
      DO 1020 I=1,N
           JI=L*(I-1)
           KK=M*(I-1)
           DO 1020 J=1,L
           JI=JI+1
           CON=0.0
           JK=J-L
           DO 1010 K=1,M
                KI=KK+K
                JK=JK+L
                CON=CON+A(JK)*B(KI)
 1010      CONTINUE
           C(JI)=CON
 1020 CONTINUE
C
      RETURN
      END


      SUBROUTINE MPYATB (A,B,C,L,M,N)
C
C  THIS PROGRAM PERFORMS THE FOLLOWING MATRIX OPERATION:
C           C(L,N) = A(M,L) TRANSPOSE * B(M,N)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1), B(1), C(1)
C
      IJ=0
      DO 1020 I=1,N
           KK=M*(I-1)
           DO 1020 J=1,L
           IK=KK
           KJ=M*(J-1)
           IJ=IJ+1
           CON=0.0
           DO 1010 K=1,M
                IK=IK+1
                KJ=KJ+1
                CON=CON+A(KJ)*B(IK)
 1010      CONTINUE
           C(IJ)=CON
 1020 CONTINUE
      RETURN
      END

      SUBROUTINE MPYABT (A,B,C,L,M,N)
```

```
C
C   THIS PROGRAM PERFORMS THE FOLLOWING MATRIX OPERATION:
C          C(L,N) = A(L,M) * B(N,M) TRANSPOSE
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1), B(1), C(1)
C
      DO 1020 I=1,N
           JI=L*(I-1)
           DO 1020 J=1,L
           JI=JI+1
           CON=0.0
           IK=I-N
           JK=J-L
           DO 1010 K=1,M
                IK=IK+N
                JK=JK+L
                CON=CON+A(JK)*B(IK)
 1010      CONTINUE
           C(JI)=CON
 1020 CONTINUE
      RETURN
      END


      SUBROUTINE ADDMAT (A,B,C,N)
C
C   THIS SUBROUTINE COMPUTES THE SUM OF TWO VECTORS
C
C   INPUT...
C     A = FIRST VECTOR.
C     B = SECOND VECTOR.
C     N = SIZE OF VECTORS A AND B.
C
C   OUTPUT...
C     C = THE SUM OF VECTORS A AND B.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1), B(1), C(1)
C
C   COMPUTE VECTORS SUM
C
      DO 1010 I=1,N
           C(I)=A(I)+B(I)
 1010 CONTINUE
      RETURN
      END


      SUBROUTINE SUBMAT (A,B,C,N)
C
C   THIS SUBROUTINE COMPUTES THE DIFFERENCE OF TWO VECTORS
C
C   INPUT...
C     A = FIRST VECTOR.
```

```
C     B = SECOND VECTOR.
C     N = SIZE OF VECTORS A AND B
C
C   OUTPUT...
C     C = THE DIFFERENCE OF VECTORS A AND B
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1), B(1), C(1)
C
      DO 1010 I=1,N
           C(I)=A(I)-B(I)
 1010 CONTINUE
      RETURN
      END


      SUBROUTINE TRANSP (A,B)
C
C   THIS SUBROUTINE TRANSPOSES THE 6X6 MATRIX A AND STORES IT IN B
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(6,6), B(6,6)
C
      DO 1010 I=1,6
           DO 1010 J=1,6
           B(I,J)=A(J,I)
 1010 CONTINUE
      RETURN
      END


      SUBROUTINE FILL (A,N,B)
C
C   THE SUBROUTINE SETS A SPECIFIED NUMBER OF SEQUENTIAL LOCATIONS TO A
C   GIVEN  VALUE.
C
C   INPUT...
C     THROUGH CALLING LIST.
C       A - THE FIRST ELEMENT OF THE SEQUENTIAL LOCATIONS.
C       N - THE NUMBER OF LOCATIONS TO BE INITIALIZED.
C       B - THE CONSTANT TO BE USED IN THE INITIALIZATION.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1)
C
      DO 1010 K=1,N
           A(K)=B
 1010 CONTINUE
      RETURN
      END
```

# PC Giant

## Source Code

## File Name: 5.FOR (Anthropometry)

## 14 June 1990

```
      SUBROUTINE STUFFP(ID1, ID2, OBJECT)
C  Search object point ids to find matching ids in anthropometry list
      REAL*8 P,  OBJECT(3)
      COMMON /TAPES/  IN,IO,IOS,IDUM(14)
      COMMON /ANTHR/P(7, 3)
c      CHARACTER*4  ID(7)
       dimension  ID(7)
      DATA ID/' lam',' ram',' lon',' ron',' ltp',' rtp',' ctp'/
      DO 20 I=1, 7
      IF (ID2.EQ.ID(I))THEN
C  Stuff object points into corresponding locations in array P
         DO 10 J=1, 3
 10      P(I, J)=OBJECT(J)
         RETURN
      ENDIF
 20   CONTINUE
C  Can't find point
C      WRITE(IO,'(/2A4,'' not in anthro list'')')ID1, ID2
C      WRITE(IOS,'(/2A4,'' not in anthro list'')')ID1, ID2
      END
C
      SUBROUTINE ANTHRO
C
C  This program verifies that we have the 7 needed anthro points,
C  calls the routine to find the transformations & prints results
C
      REAL*8 P, X(3),  AB(3, 3)
      COMMON /TAPES/  IN,IO,IOS,IDUM(14)
      COMMON /ANTHR/P(7, 3)
C
      CALL NEWPAG
      WRITE(IO,'(44XA40//)')'A N T H R O P O M E T R Y   O U T P U T'
      WRITE(IOS,'(20XA40//)')'A N T H R O P O M E T R Y   O U T P U T'
      DO 10 I=1, 7
      IF(P(I,3).EQ.0)THEN
         WRITE(IO, *)' Can''t find 7 non-zero anthro points--halting'
         WRITE(IOS, *)' Can''t find 7 non-zero anthro points--halting'
         RETURN
      ENDIF
 10   CONTINUE
      CALL NBDL (X, AB)
      WRITE(IO, 8)X, AB
      WRITE(IOS, 9)X, AB
  8   FORMAT(38X'T-PLATE ORIGIN WITH RESPECT TO HEAD ANATOMICAL ORIGIN'
     . //41X'X= '2PF8.4,'cm     Y= 'F8.4,'cm     Z= 'F8.4,'cm'0P///
     . 35X'T-PLATE ORIENTATION WITH RESPECT TO HEAD ANATOMICAL SYSTEM'
     . //3(47X,3F11.6/))
  9   FORMAT(14X'T-PLATE ORIGIN WITH RESPECT TO HEAD ANATOMICAL ORIGIN'
     . //17X'X= '2PF8.4,'cm     Y= 'F8.4,'cm     Z= 'F8.4,'cm'0P///
     . 11X'T-PLATE ORIENTATION WITH RESPECT TO HEAD ANATOMICAL SYSTEM'
     . //3(23X,3F11.6/))
      END

      SUBROUTINE UVEC (A,K)
C
```

```
C   THIS PROGRAM PERFORMS THE FOLLOWING MATRIX OPERATION:
C             A(K,-) = A(K,-) / MAGNITUDE (A(K, -))
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(3, 3)
C
      B=0
      DO 10 I=1, 3
 10   B=B+A(K, I)**2
      B=DSQRT(B)
      DO 20 I=1, 3
 20   A(K, I)=A(K, I)/B
      END
C
C
      SUBROUTINE NBDL (X, AB)
C
C   THIS PROGRAM FINDS THE ORIGIN & TRANSFORMATION MATRIX OF THE
C   T-PLATE RELATIVE TO THE HEAD ANATOMICAL ORIGIN IN THE HEAD
C   ANATOMICAL COORDINATE SYSTEM
C
      COMMON /ANTHR/P(7, 3)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION  X(3),  AB(3, 3),  A(3, 3), B(3, 3), Q(3)
C
      DO 10 I=1, 3
C   Find origin of Head Anatomical Coordinate System
      X(I)=(P(1,I)+P(2,I))/2
C   Find x-axis
      A(1,I)=(P(3,I)+P(4,I))/2-X(I)
C   Find origin of T-plate
      Q(I)=P(7,I)
C   Find x-axis of T-plate
      B(1,I)=(P(5,I)+P(6,I))/2-Q(I)
C   Find approx. y-axes
      A(2,I)=P(1,I)-X(I)
      B(2,I)=P(6,I)-P(5,I)
C   Find vector from head anat to T-plate
 10   Q(I)=Q(I)-X(I)
C   Make unit vectors of x-axes
      CALL UVEC(A, 1)
      CALL UVEC(B, 1)
C   Find components of the approx y-axes along the respective x-axes
      DO 20 I=1, 3
      C=C+A(1,I)*A(2,I)
 20   D=D+B(1,I)*B(2,I)
C   Subtract these to yield y-axes perpendicular to the resp x-axes
      DO 30 I=1, 3
      A(2,I)=A(2,I)-C*A(1,I)
 30   B(2,I)=B(2,I)-D*B(1,I)
C   Make them of unit length
      CALL UVEC(A, 2)
      CALL UVEC(B, 2)
C   Find the z-axes by taking the cross products of the x-axes & y-axes
      DO 40 I=1, 3
```

```
      J=I+1
      IF(J.GT.3)J=J-3
      K=I+2
      IF(K.GT.3)K=K-3
      A(3,I)=A(1,J)*A(2,K)-A(1,K)*A(2,J)
 40   B(3,I)=B(1,J)*B(2,K)-B(1,K)*B(2,J)
C  Find the components of the transformation vector and matrix in
C  the head anatomical coordinate system
      DO 50 I=1, 3
      X(I)=0.D0
      DO 50 J=1, 3
      X(I)=X(I)+Q(J)*A(I,J)
      AB(I,J)=0.D0
      DO 50 K=1, 3
 50   AB(I,J)=AB(I,J)+B(I,K)*A(J,K)
      END
```

# PC Giant

## Source Code

## File Name: various.INC

## (GIANT Common Statement Include Files)

## 14 June 1990

```
File Name: COEFF.INC
      COMMON /COEFF/  A(2,3),C(2),B(2,6)

File Name: CONVCR.INC
      COMMON /CONVCR/ EPSLN ,IRESA ,NIT

File Name: EARTHD.INC
      COMMON /EARTHD/ SPHRD(2)

File Name: FORMTS.INC
C  Set output card format for camera parameters and triangulated points
      CHARACTER*19 IOFM1, IOFM2
      DATA IOFM1/'(2A4,3F12.3,3G10.4)'/
      DATA IOFM2/'(2A4,3F12.3,3G10.4)'/

File Name: GIANT.INC
      COMMON /PAGEN/ IPAGE

File Name: GPCTRS.INC
      COMMON /GPCTRS/ NGPS,NIND

File Name: HPUNIX.INC
      COMMON /HPUNIX/ NB

File Name: INDXFR.INC
      COMMON /INDXFR/ INDEXM(3,ISZ1),IBUF(400)

File Name: OPTION.INC
      COMMON /OPTION/ IUNIT ,IATT  ,ILTGP ,IPNGP,ILTST ,IPNST

File Name: OPTON2.INC
      COMMON /OPTON2/ ITRNG ,IPROP ,IWGHT ,ISORT,NCNTRL,IEIGEN

File Name: OPTON4.INC
      COMMON /OPTON4/ IAREFR,IWREFR,WLEVEL,CNW

File Name: PAGEN.INC
      COMMON /PAGEN/ IPAGE

File Name: RANVAR.INC
      COMMON /RANVAR/ IP

File Name: ROTAT.INC
      COMMON /ROTAT/  R(3,3,ISZ6),PR(3,3,ISZ6),PQ(3,2,ISZ6),
     .                RL(3,3,ISZ6),STATON(3,ISZ6),DSTATN(3,3,ISZ6)

File Name: SWITCH.INC
      COMMON /SWITCH/ IS

File Name: TAPES.INC
      INTEGER CAMERA,FRAMES,OBJECT
      COMMON /TAPES/      IN,IO,IOS,   IP1,   IP2,
     .                CAMERA,IMAGES,FRAMES,OBJECT,
     .                ITAPE1,ITAPE2,ITAPE3,ITAPE4,
     .                ITAPE5,ITAPE6,ITAPE7,ITAPE0

File Name: TITLEP.INC
      COMMON /TITLEP/ JTITLE(20)

File Name: UNITVR.INC
      COMMON /UNITVR/ SS,IDFREE
```

```
   File Name: WARNGS.INC
         COMMON /WARNGS/ INPCTR

File Name: WORK11.INC
         REAL*4              VARPLT,FOCAL,WTMAT
         COMMON /WORK11/ PARAM(6,ISZ1),VARPLT(2,ISZ1),FOCAL(ISZ1),
     .             WTMAT(6,ISZ1),IDCAM(2,ISZ1),INDEX(2,ISZ1),IDPLT(2,ISZ2)
     .            ,  idddum(17, isz1)

File Name: WORK21.INC
         REAL*4              ACCSOL,VARPLT,FOCAL,WTMAT
         COMMON /WORK21/ PARAM(6,ISZ1),SOLUTM(6,ISZ1),ACCSOL(6,ISZ1),
     .                   VARPLT(2,ISZ1),FOCAL(ISZ1),WTMAT(6,ISZ1),
     .                   IFOTO(2,ISZ1),NCAM

File Name: WORK22.INC
         COMMON /WORK22/ EQN(ISZ8),CONV(ISZ9),TMPST(36,ISZ6)

File Name: WORK24.INC
         COMMON /WORK24/ IDCAM(ISZ1),IDS(ISZ1),NMAX

File Name: WORK25.INC
         COMMON /WORK25/ R(3,3,ISZ1),STATON(3,ISZ1),RL(3,3,ISZ1)

File Name: PARAMS.INC
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C  *                          EXAMPLE DIMENSIONS:                         *
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C                                           VAX     PC     PC     PC     PC     PC
C                                        11/750   640K   640K   512K   512K   DEMO
C                                                  wo/     w/    wo/     w/    wo/
C                                                 8087   8087   8087   8087   8087
C
C  MAX Camera Stations [N1]   (ISZ1)      450     100    150     26     37      6
C
C  MAX Object Points          (ISZ2)    10000    2000   3000    520    740     40
C
C  MAX Control Points [>= N1](ISZ3)       450      90    140     25     36      5
C
C  MAX Frames a Unique Point
C                 Appears On (ISZ4)        10      10     10     10     10      6
C
C  MAX Camera Systems         (ISZ5)       10      10     10     10     10      2
C
C  Normal Equations
C              Band Width [N](ISZ6)        23      23     23     23     23      6
C
C  Reduced Normal Equations
C                    [N - 1](ISZ7)         22      22     22     22     22      5
C
C  Size of Coefficient Matrix
C    {[ N (N + 1) / 2 ] * 36}(ISZ8)      9936    9936   9936   9936   9936    756
C
C  Size of Constant Vector
C                    [N * 6](ISZ9)        138     138    138    138    138     36
C
         PARAMETER      (ISZ1 = 26, ISZ2 =   520, ISZ3 = 25,
     .                   ISZ4 = 10, ISZ5 =    10, ISZ6 = 23,
     .                   ISZ7 = 22, ISZ8 =  9936, ISZ9 =138)
```

# PC Giant

## Source Code

## File Name: PREP.FOR

## (Pre-processor Program For GIANT)

14 June 1990

```
      PROGRAM MAIN
C
C $CONFIG$="/T1 /LC"
C $NAME$
C       MAIN
C $PATHS$
C       FUNCTIONS\ALL
C       MODULES\MAIN
C $1$
C
C  Input data for the Preprocessing Program:
C
C  OPTION CARD:
C
C     3 in col. 1  Three-parameter transformation
C     4 in col. 1  Four-parameter transformation
C     5 in col. 1  Five-parameter transformation
C     6 in col. 1  Six-parameter transformation
C     8 in col. 1  Eight-parameter transformation
C     0 in col. 2  means do not correct for atmospheric refraction
C     1 in col. 2  means correct for atmospheric refraction
C     1 in col. 3  means to multiply input by 25.4 (inches to mm)
C
C  CALIBRATED FIDUCIAL CARDS:
C     Calibrated Fiducial Coordinates in   FORMAT (2X,I4,4X,2F10.4)
C
C  END OF CALIBRATED FIDUCIAL MARKER:
C     0 in COLUMNS 1-10
C     Radial Lens Distortion functions in  FORMAT (3E10.5/3E10.5)
C     Decent Lens Distortion functions in  FORMAT (3E10.5)
C     Atmospheric Refraction # of entries  FORMAT (I2)
C
C  IF PREVIOUS CARD HAD A NUMBER GREATER THAN ZERO:
C     Atmospheric Refraction data in table FORMAT (2F10.3)
C
C  REPEAT FOR EACH FRAME MEASURED:
C *********************************************************************
C  MEASURED DATA SET:
C     Frame IDentification in              FORMAT (A8)
C     Observed Fiducial Coordinates in     FORMAT (6X,I4,6F10.3)
C
C  BLANK CARD
C     Observed Plate Coordinates in        FORMAT (2X,A8,6F10.3)
C *********************************************************************
C  END OF JOB CARD:
C     ********** (ASTERISKS IN COLUMNS 1-10.)
C
C                                                        $SKIP START$
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      CHARACTER*8 IDPT,IFRAM,IBLANK,IEND
      DIMENSION DISTM(2,50), TEMPM1(2,5), CALC(2,2000), IDFD(50)
      COMMON CALCOR(2,50),OBSCOR(2,50),EQN(8,9),DEL(8),ICH3, NFID
      DATA IBLANK/'        '/, IEND/'********'/
C
      OPEN (UNIT=7,FILE='PREP.IN')
```

```
      OPEN (UNIT=8,FILE='PREP.OUT',CARRIAGE CONTROL='FORTRAN')
      OPEN (UNIT=10,FILE='PREP80.OUT',CARRIAGE CONTROL='FORTRAN')
      OPEN (UNIT=9,FILE='IMAGES.OUT')
C                                                            $SKIP END$
C  Read order of transformation & 1 for atmospheric refraction
C                                                          $SKIP START$
      READ (7,1370) IOPT1,IOPT2,IOPT3
      IF (IOPT1.GT.6)IOPT1=8
      ICH3=0
      IF (IOPT1.LE.3) THEN
           ICH3=1
           IOPT1=3
      END IF
      FACT=1.0D0
      IF(IOPT3.NE.0)FACT=25.4D0
C                                                            $SKIP END$
C  IOPT4<>0 causes sign change in 'X'
C                                                          $SKIP START$
      IOPT4=0
C                                                            $SKIP END$
C  NRED indicates the number of replications of plate coordinates
C                                                          $SKIP START$
      NRED=1
      WRITE (8,1380)
      WRITE (10,2380)
      WRITE (8,1400)
      WRITE (10,2400)
C                                                            $SKIP END$
C  Read Calibrated Fiducial Coordinates
C                                                          $SKIP START$
      NFID=0
 1010 READ (7,1410) IFID,X,Y
         X=X*FACT
         Y=Y*FACT
      MAXFID=MAXFID+1
      IF (IFID.EQ.0) GO TO 1030
      IF (IFID.GT.2000) GO TO 1020
      CALC(1,IFID)=X
      CALC(2,IFID)=Y
      WRITE (8,1420) IFID,X,Y
      WRITE (10,2420) IFID,X,Y
      GO TO 1010
 1020 WRITE (8,*) '0 ILLEGAL MASTER FIDUCIAL ID'
      WRITE (*,1430) IFID
      STOP
C                                                            $SKIP END$
C  Read And List Lens Distortion Parameters.
C                                                          $SKIP START$
 1030 READ (7,1440) FK1,FK2,FK3, FK4,FK5,FK6, FJ1,FJ2,PHIO
      IF (FJ1+FJ2+PHIO.EQ.0) THEN
           WRITE (8,1460) FK1,FK2,FK3,FK4,FK5,FK6
           WRITE (10,2460) FK1,FK2,FK3,FK4,FK5,FK6
      ELSE
           WRITE (8,1460) FK1,FK2,FK3,FK4,FK5,FK6,FJ1,FJ2,PHIO
           WRITE (10,2460) FK1,FK2,FK3,FK4,FK5,FK6,FJ1,FJ2,PHIO
```

```
      END IF
      SINPHI=DSIN(PHIO)
      COSPHI=DCOS(PHIO)
C                                                                 $SKIP END$
C  Read And List Atmospheric Refraction Table.
C                                                                 $SKIP START$
      READ (7,1470) NINT
      IF(NINT.GT.0) READ (7,1480)  ((DISTM(I,J),I=1,2),J=1,NINT)
      IF (IOPT2.EQ.0) GO TO 1050
      WRITE (8,1490)
      WRITE (10,2490)
      DO 1040 I=1,NINT
           WRITE (8,1500) DISTM(1,I),DISTM(2,I)
           WRITE (10,2500) DISTM(1,I),DISTM(2,I)
 1040 CONTINUE
C                                                                 $SKIP END$
C  Read & Write Frame ID
C                                                                 $SKIP START$
 1050 READ  (7,1515) IFRAM
      WRITE (9,1515) IFRAM
      IF(IFRAM.EQ.IEND)STOP
      WRITE (8,1380)
      WRITE (10,2380)
      WRITE (8,1520) IFRAM
      WRITE (10,2520)  IFRAM
      WRITE (8,1530)
      WRITE (10,2530)
      DO 1170 K=1,MAXFID
C                                                                 $SKIP END$
C  Read measured fiducial coordinates
C                                                                 $SKIP START$
           READ (7,1510) KK,((TEMPM1(I,J),I=1,2),J=1,NRED)
           DO 1060 I=1, 2
           DO 1060 J=1, NRED
 1060      TEMPM1(I, J)=TEMPM1(I, J)*FACT
           IF (KK.EQ.0) GO TO 1180
 1090      XMAX=0.0D0
           YMAX=0.0D0
           XMIN=1000.0D0
           YMIN=1000.0D0
           SUMX=0.0D0
           SUMY=0.0D0
           DO 1120 J=1,NRED
                X=TEMPM1(1,J)
                Y=TEMPM1(2,J)
                IF (X.EQ.0.AND.Y.EQ.0) GO TO 1130
                SUMX=SUMX+X
                SUMY=SUMY+Y
                IF (NRED.EQ.1) GO TO 1120
                IF (XMAX.LT.X) XMAX=X
                IF (XMIN.GT.X) XMIN=X
                IF (YMAX.LT.Y) YMAX=Y
                IF (YMIN.GT.Y) YMIN=Y
 1120      CONTINUE
           IF (NRED.NE.1) GO TO 1140
```

```
 1130       XMIN=0.0D0
            YMIN=0.0D0
 1140       J=NRED
            IF (J.EQ.0) J=1
            XT=SUMX/J
            YT=SUMY/J
            IF (IOPT4.NE.0) XT=-XT
            OBSCOR(1,K)=XT
            OBSCOR(2,K)=YT
            CALCOR(1,K)=CALC(1,KK)
            CALCOR(2,K)=CALC(2,KK)
            IDFD(K)=KK
            X=XMAX-XMIN
            Y=YMAX-YMIN
            WRITE (8,1540) KK,XT,YT,X,Y
            WRITE (10,2540) KK,XT,YT,X,Y
 1170 CONTINUE
C                                                              $SKIP END$
C  Compute the Multi-Parameter Transformation.
C                                                            $SKIP START$
 1180 NFID=K-1
      IF (IOPT1.LE.5) CALL FOURP
      IF (IOPT1.EQ.5) CALL FIVEP
      IF (IOPT1.EQ.6) CALL SIXP
      IF (IOPT1.EQ.8) CALL EIGHTP
 1230 WRITE (8,1550) IOPT1
   \  WRITE (10,2550) IOPT1
C                                                              $SKIP END$
C  Compute Residuals For the Fiducial Coordinates
C                                                            $SKIP START$
      DO 1240 I=1,NFID
        X=OBSCOR(1,I)
        Y=OBSCOR(2,I)
       XT=(X*DEL(1)+Y*DEL(2)+DEL(3))/(X*DEL(4)+Y*DEL(5)+1.0)-CALCOR(1,I)
       YT=(X*DEL(6)+Y*DEL(7)+DEL(8))/(X*DEL(4)+Y*DEL(5)+1.0)-CALCOR(2,I)
        KK=IDFD(I)
        WRITE (8,1560) KK,XT,YT
        WRITE (10,2560) KK,XT,YT
 1240 CONTINUE
      IF (NRED .GT. 1) WRITE (8,1570)
      IF (NRED .GT. 1) WRITE (10,2570)
      IF (NRED .EQ. 1) WRITE (8,1575)
      IF (NRED .EQ. 1) WRITE (10,2575)
C                                                              $SKIP END$
C  Compute the Averaged Coordinates of the Measured Control Points
C                                                            $SKIP START$
 1250 READ (7,1580) IDPT,((TEMPM1(I,J),I=1,2),J=1,NRED)
            DO 1255 I=1, 2
            DO 1255 J=1, NRED
 1255       TEMPM1(I, J)=TEMPM1(I, J)*FACT
      IF (IDPT.NE.IBLANK) GO TO 1260
      WRITE (9,*) '********'
      GO TO 1050
 1260 XMAX=0.0D0
      YMAX=0.0D0
```

```
      XMIN=1000.0D0
      YMIN=1000.0D0
      SUMX=0.0D0
      SUMY=0.0D0
      DO 1290 J=1,NRED
           X=TEMPM1(1,J)
           Y=TEMPM1(2,J)
           IF (X.EQ.0.AND.Y.EQ.0) GO TO 1310
           SUMX=SUMX+X
           SUMY=SUMY+Y
           IF (NRED.EQ.1) GO TO 1290
           IF (XMAX.LT.X) XMAX=X
           IF (XMIN.GT.X) XMIN=X
           IF (YMAX.LT.Y) YMAX=Y
           IF (YMIN.GT.Y) YMIN=Y
 1290 CONTINUE
      IF (NRED.NE.1) GO TO 1300
      XMIN=0.0D0
      YMIN=0.0D0
 1300 J=NRED+1
 1310 J=J-1
      IF (J.EQ.0) GO TO 1050
      XM=XMAX-XMIN
      YM=YMAX-YMIN
      X=SUMX/J
      Y=SUMY/J
      IF (IOPT4.NE.0) X=-X
C                                                              $SKIP END$
C  Correct Measured Coordinates for Film Shrinkage
C                                                            $SKIP START$
      XT=(X*DEL(1)+Y*DEL(2)+DEL(3))/(X*DEL(4)+Y*DEL(5)+1.0)
      YT=(X*DEL(6)+Y*DEL(7)+DEL(8))/(X*DEL(4)+Y*DEL(5)+1.0)
C                                                              $SKIP END$
C  Correct for Radial Lens Distortion:
C                                                            $SKIP START$
      RT2=(XT**2+YT**2)
      RT4=RT2*RT2
      RT6=RT4*RT2
      C1=FK1*RT2+FK2*RT4+FK3*RT6+1.
C                                                              $SKIP END$
C  Correct for Tangential Lens Distortion:
C                                                            $SKIP START$
      C2=FJ1*RT2+FJ2*RT4
      XT=C1*XT-C2*SINPHI
      YT=C1*YT+C2*COSPHI
C                                                              $SKIP END$
C  Correct Measured Coordinates for Atmospheric Refraction:
C                                                            $SKIP START$
      RT=DSQRT(XT**2+YT**2)
      DO 1330 II=1,NINT
           IF (RT.LE.DISTM(1,II)) GO TO 1340
 1330 CONTINUE
      IF (IOPT2.EQ.0) GO TO 1350
      WRITE (8,1590) IDPT
      GO TO 1350
```

```
1340 DR=DISTM(2,II)-(DISTM(2,II-1)-DISTM(2,II)) /
    .                        DISTM(1,II-1-DISTM(1,II))*(DISTM(1,II)-RT)
     XT=DR/RT*XT+XT
     YT=DR/RT*YT+YT
1350 IF (NRED .GT. 1) WRITE (8,1600) IDPT,X,Y,XT,YT,XM,YM
     IF (NRED .GT. 1) WRITE (10,2600) IDPT,X,Y,XT,YT,XM,YM
     IF (NRED .EQ. 1) WRITE (8,1605) IDPT,X,Y,XT,YT
     IF (NRED .EQ. 1) WRITE (10,2605) IDPT,X,Y,XT,YT
C                                                     $SKIP END$
C  Write Records for Aerotriangulation Input:
C                                                     $SKIP START$
     WRITE (9,1610) IDPT,XT,YT,IFRAM
     GO TO 1250
C
1370 FORMAT (3I1)
1380 FORMAT ('1', 43X, 'PC GIANT PREPROCESSOR  JUNE 1990'/)
1400 FORMAT (45X,31HCALIBRATED FIDUCIAL COORDINATES)
1410 FORMAT (2X,I4,4X,2F10.4)
1420 FORMAT (45X,I4,5X,F8.3,5X,F8.3)
1430 FORMAT (/////,I20,' EXCEEDS THE MAXIMUM OF 2000 FIDUCIALS')
1440 FORMAT (3E10.5)
1460 FORMAT (/51X,'LENS DISTORTION'//51X,'RADIAL PARAMETERS'/31X'K1='
    .E15.8,5H  K2=E15.8,5H  K3=E15.8/31X'K4='E15.8,5H  K5=E15.8,5H  K6=
    .E15.8//:45X,28HLENS DECENTRATION PARAMETERS/31X,3HJ1=F15.8,5H  J2=
    .E15.8,5H PHI=E15.8/)
1470 FORMAT (I2)
1480 FORMAT (2F10.3)
1490 FORMAT (40X,39HATMOSPHERIC REFRACTION DISTORTION TABLE)
1500 FORMAT (44X,2F13.3)
1510 FORMAT (6X,I4,6F10.3)
1515 FORMAT (A8)
1520 FORMAT (40X,'FIDUCIAL MEASUREMENTS OF FRAME ',A8//)
1530 FORMAT (36X,2HID,12X,7HAVERAGE,13X,10HMAX SPREAD/48X,'X',9X,'Y',
    .  11X,'X',9X,'Y')
1540 FORMAT (36X,I4,2X,2F10.3,2X,2F10.3)
1550 FORMAT (//I38,'-PARAMETER RESIDUALS OF THE FIDUCIAL COORDINATES'/)
1560 FORMAT (42X,I4,2F15.3)
1570 FORMAT (//52X,17HPLATE COORDINATES//22X,2HID,11X,8HMEASURED,13X,8H
    .ADJUSTED,13X,10HMAX SPREAD,11X,5HFRAME/34X,'X',9X,'Y',10X,'X',9X,
    . 'Y',10X,'X',9X,'Y')
1575 FORMAT (//52X,17HPLATE COORDINATES//38X,2HID,11X,8HMEASURED,13X,8H
    .ADJUSTED/50X,'X',9X,'Y',10X,'X',9X,'Y')
1580 FORMAT (2X,A8,6F10.3)
1590 FORMAT (' POINT ', A8, ' WAS NOT CORRECTED FOR LENS DISTORTION AND
    . ATMOSPHERIC REFRACTION'//)
1600 FORMAT (18X,A8,2X,2F10.3,1X,2F10.3,1X,2F10.3)
1605 FORMAT (34X,A8,2X,2F10.3,1X,2F10.3)
1610 FORMAT (A8,2X,2F10.4,'  Photo ',A8)
C 80 col
2380 FORMAT ('1', 23X, 'PC Giant Preprocessor  June 1990'/)
2400 FORMAT (25X,31HCalibrated Fiducial Coordinates)
2420 FORMAT (25X,I4,5X,F8.3,5X,F8.3)
2460 FORMAT (/31X,'Lens Distortion'//31X,'Radial Parameters'/11X'K1='
    .E15.8,5H  K2=E15.8,5H  K3=E15.8/11X'K4='E15.8,5H  K5=E15.8,5H  K6=
    .E15.8//:25X,28HLens Decentration Parameters/11X,3HJ1=F15.8,5H  J2=
```

```
     .E15.8,5H PHI=E15.8/)
 2490 FORMAT (20X,39HAtmospheric Refraction Distortion Table)
 2500 FORMAT (24X,2F13.3)
 2520 FORMAT (20X,'Fiducial Measurements of Frame ',A8//)
 2530 FORMAT (16X,2HID,12X,7HAverage,13X,10HMax Spread/28X,'X',9X,'Y',
     .   11X,'X',9X,'Y')
 2540 FORMAT (16X,I4,2X,2F10.3,2X,2F10.3)
 2550 FORMAT (//I18,'-Parameter Residuals of the Fiducial Coordinates'/)
 2560 FORMAT (22X,I4,2F15.3)
 2570 FORMAT (//32X,17HPLATE COORDINATES//'      ID',11X,8HMeasured,13X,
     .'Adjusted',13X,'Max Spread',11X,'Frame'/17X,'X',9X,'Y',10X,'X',9X,
     . 'Y',10X,'X',9X,'Y')
 2575 FORMAT (//32X,17HPLATE COORDINATES//18X,2HID,11X,8HMeasured,13X,8H
     .Adjusted/30X,'X',9X,'Y',10X,'X',9X,'Y')
 2600 FORMAT (X,A8,2X,2F10.3,1X,2F10.3,1X,2F10.3)
 2605 FORMAT (14X,A8,2X,2F10.3,1X,2F10.3)
C                                                              $SKIP END$
C                                                                  $END$
      END
C***********************************************************************
      SUBROUTINE FOURP
C
C $CONFIG$="/T1 /LC"
C $NAME$
C           SUBROUTINE FOURP
C $PATHS$
C  \        FUNCTIONS\ALL
C           MODULES\FOURP
C $1$
C
C  Calculate the 3 or 4 Parameter Transformation Between an Exact Set
C       of Data and a Corresponding Set of Measured Data.
C
C                                                            $SKIP START$
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION AM(2,4), CM(2)
      COMMON CALCOR(2,50),OBSCOR(2,50),EQN(8,9),DEL(8),ICH3, NFID
C
      DO 1010 I=1,4
           DO 1010 J=1,5
           EQN(I,J)=0.0D0
 1010 CONTINUE
      AM(1,3)=1.0D0
      AM(1,4)=0.0D0
      AM(2,3)=0.0D0
      AM(2,4)=1.0D0
      DO 1030 I=1,NFID
           AM(1,1)=OBSCOR(1,I)
           AM(1,2)=OBSCOR(2,I)
           AM(2,1)=AM(1,2)
           AM(2,2)=-AM(1,1)
           CM(1)=CALCOR(1,I)
           CM(2)=CALCOR(2,I)
           DO 1020 J=1,4
                DO 1020 K=1,2
```

```
               EQN(J,5)=EQN(J,5)+AM(K,J)*CM(K)
               DO 1020 L=1,4
               EQN(J,L)=EQN(J,L)+AM(K,J)*AM(K,L)
 1020      CONTINUE
 1030 CONTINUE
      CALL LINSOL(4)
      IF (ICH3.EQ.0) GO TO 1060
C                                                           $SKIP END$
C  If ICH3<>0 Transform the 4-param to a 3-param
C                                                         $SKIP START$
      SCALE=EQN(1,5)**2+EQN(2,5)**2
      SCALE=DSQRT(SCALE)
      EQN(1,5)=EQN(1,5)/SCALE
      EQN(2,5)=EQN(2,5)/SCALE
      SUM1=0.0D0
      SUM2=0.0D0
      DO 1050 I=1,NFID
           X=OBSCOR(1,I)
           Y=OBSCOR(2,I)
           SUM1=SUM1+CALCOR(1,I)-EQN(1,5)*X-EQN(2,5)*Y
           SUM2=SUM2+CALCOR(2,I)+EQN(2,5)*X-EQN(1,5)*Y
 1050 CONTINUE
      EQN(3,5)=SUM1/NFID
      EQN(4,5)=SUM2/NFID
C                                                           $SKIP END$
C  Form transformation parameters vector
C  \                                                      $SKIP START$
 1060 DEL(1)=EQN(1,5)
      DEL(2)=EQN(2,5)
      DEL(3)=EQN(3,5)
      DEL(4)=0.0D0
      DEL(5)=0.0D0
      DEL(6)=-DEL(2)
      DEL(7)=DEL(1)
      DEL(8)=EQN(4,5)
      RETURN
C                                                           $SKIP END$
C                                                                $END$
      END
C**********
      SUBROUTINE FIVEP
C
C $CONFIG$="/T1 /LC"
C $NAME$
C          SUBROUTINE FIVEP
C $PATHS$
C          FUNCTIONS\ALL
C          MODULES\FIVEP
C $1$
C
C  Calculate the FIVE Parameter Transformation Between an Exact Set
C       of Data and a Corresponding Set of Measured Data.
C
C                                                         $SKIP START$
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
```

```
      DIMENSION B(2,5),C(2),CV(5),PAR(5)
      COMMON CALCOR(2,50),OBSCOR(2,50),EQN(8,9),DEL(8),ICH3, NFID
C
      PAR(1)=DSQRT(DEL(1)**2+DEL(2)**2)
      PAR(2)=PAR(1)
      PAR(3)=DATAN2(DEL(2),DEL(1))
      PAR(4)=DEL(3)
      PAR(5)=DEL(8)
      B(1,2)=0.0D0
      B(1,5)=0.0D0
      B(2,1)=0.0D0
      B(2,4)=0.0D0
      DO 30 II=1,10
      DO 2 I=1, 5
        CV(I)=0.0D0
        DO 2 J=1, 5
 2    EQN(I, J)=0.0D0
      DO 10 I=1,NFID
           B(1,4)=PAR(1)
           B(2,5)=PAR(2)
           SINT=DSIN(PAR(3))
           COST=DCOS(PAR(3))
           X=OBSCOR(1,I)
           Y=OBSCOR(2,I)
           C1=-X*SINT+Y*COST
           C2= X*COST+Y*SINT
           B(1,1)=C2*PAR(1)
           B(1,3)=C1*PAR(1)**2
           B(2,2)=C1*PAR(2)
           B(2,3)=-C2*PAR(2)**2
           C(1)=PAR(1)*(CALCOR(1,I)-PAR(1)*C2-PAR(4))
           C(2)=PAR(2)*(CALCOR(2,I)-PAR(2)*C1-PAR(5))
           DO 10 J=1,5
           DO 10 K=1,2
                CV(J)=CV(J)+B(K,J)*C(K)
           DO 10 L=1,5
                EQN(J,L)=EQN(J,L)+B(K,J)*B(K,L)
 10        CONTINUE
C                                                              $SKIP END$
C    Solve normal equations
C                                                            $SKIP START$
      CALL LINSOL(5)
      DO 15 J=1, 5
 15   PAR(J)=PAR(J)+EQN(J, 6)
C                                                              $SKIP END$
C    Test for convergence
C                                                            $SKIP START$
      DO 20 J=1, 5
      C1=DABS(EQN(J, 6))
      EPSLN=1.0D-6
      IF(J.GT.3)EPSLN=1.0D-4
      IF(C1.GT.EPSLN)GO TO 30
 20   CONTINUE
      GO TO 40
 30   CONTINUE
```

```
      WRITE(*,*)' Error in FIVE'
C                                                               $SKIP END$
C   Form transformation parameters vector
C                                                             $SKIP START$
 40   SINT=DSIN(PAR(3))
      COST=DCOS(PAR(3))
      DEL(1)=PAR(1)*COST
      DEL(2)=PAR(1)*SINT
      DEL(3)=PAR(4)
      DEL(4)=0.0D0
      DEL(5)=0.0D0
      DEL(6)=-PAR(2)*SINT
      DEL(7)=PAR(2)*COST
      DEL(8)=PAR(5)
      RETURN
C                                                               $SKIP END$
C                                                                    $END$
      END
C**********
      SUBROUTINE SIXP
C
C $CONFIG$="/T1 /LC"
C $NAME$
C         SUBROUTINE SIXP
C $PATHS$
C         FUNCTIONS\ALL
C  \      MODULES\SIXP
C $1$
C
C  Calculate the SIX Parameter Transformation Between an Exact Set
C       of Data and a Corresponding Set of Measured Data.
C
C                                                             $SKIP START$
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION ANS(2,3), CCC(3,3), DDD(3,2), RRR(2,2), LLL(3), MMM(3),
     .          ERR(2)
      COMMON CALCOR(2,50),OBSCOR(2,50),EQN(8,9),DEL(8),ICH3, NFID
C                                                               $SKIP END$
C  Zero Normal Equation Area.
C                                                             $SKIP START$
      DO 1010 I=1,2
           DO 1010 J=1,3
           CCC(I,J)=0.0D0
           DDD(J,I)=0.0D0
 1010 CONTINUE
C                                                               $SKIP END$
C  Compute Normal Equations
C                                                             $SKIP START$
      DO 1020 I=1,NFID
           DO 1020 J=1,2
           CCC(J,3)=CCC(J,3)+CALCOR(J,I)
           DDD(3,J)=DDD(3,J)+OBSCOR(J,I)
           DO 1020 K=1,2
           CCC(J,K)=CCC(J,K)+CALCOR(J,I)*CALCOR(K,I)
           DDD(J,K)=DDD(J,K)+CALCOR(J,I)*OBSCOR(K,I)
```

```
 1020 CONTINUE
      CCC(3,1)=CCC(1,3)
      CCC(3,2)=CCC(2,3)
      CCC(3,3)=NFID
C                                                                  $SKIP END$
C  Compute Inverse of Normal Matrix.
C                                                                $SKIP START$
      IGGY=3
      CALL INVERT (CCC,IGGY,DET,LLL,MMM)
C                                                                  $SKIP END$
C  Compute the Transformation Parameters
C                                                                $SKIP START$
      DO 1030 I=1,2
           DO 1030 J=1,3
           ANS(I,J)=0.0D0
           DO 1030 K=1,3
 1030      ANS(I,J)=ANS(I,J)+CCC(J,K)*DDD(K,I)
C                                                                  $SKIP END$
C  Calculate the Transformation from Measured Data to Exact Data.
C                                                                $SKIP START$
      DO 1040 I=1,2
           DO 1040 J=1,2
 1040      RRR(I,J)=ANS(I,J)
      IGGY=2
      CALL INVERT (RRR,IGGY,DET,LLL,MMM)
      DO 1050 I=1,2
           DO 1050 J=1,2
 1050      ANS(I,J)=RRR(I,J)
      DO 1060 I=1,2
 1060      ERR(I)=-ANS(I,1)*ANS(1,3)-ANS(I,2)*ANS(2,3)
      DO 1070 I=1,2
 1070      ANS(I,3)=ERR(I)
C                                                                  $SKIP END$
C  Form transformation parameters vector
C                                                                $SKIP START$
      DEL(1)=ANS(1,1)
      DEL(2)=ANS(1,2)
      DEL(3)=ANS(1,3)
      DEL(4)=0.0D0
      DEL(5)=0.0D0
      DEL(6)=ANS(2,1)
      DEL(7)=ANS(2,2)
      DEL(8)=ANS(2,3)
      RETURN
C                                                                  $SKIP END$
C                                                                       $END$
      END
C**********
      SUBROUTINE EIGHTP
C
C $CONFIG$="/T1 /LC"
C $NAME$
C         SUBROUTINE EIGHTP
C $PATHS$
C         FUNCTIONS\ALL
```

```
C         MODULES\EIGHTP
C $1$
C
C  Calculate the EIGHT Parameter Transformation Between an Exact Set
C       of Data and a Corresponding Set of Measured Data.
C
C                                                              $SKIP START$
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON CALCOR(2,50),OBSCOR(2,50),EQN(8,9),DEL(8),ICH3, NFID
C                                                                $SKIP END$
C  Zero the matrix of linear equations EQN
C                                                              $SKIP START$
      DO 1010 I=1,8
           DO 1010 J=1,9
 1010      EQN(I,J)=0.0D0
C                                                                $SKIP END$
C  Compute approximate values for the transformation parameters
C                                                              $SKIP START$
      DO 1020 I=1,NFID
 1020 CALL ACCAPR (CALCOR(1,I),CALCOR(2,I),OBSCOR(1,I),OBSCOR(2,I))
      CALL LINSOL(8)
      DO 1030 I=1,8
 1030 DEL(I)=EQN(I,9)
C                                                                $SKIP END$
C  Compute the transformation parameters by least squares
C                                                              $SKIP START$
  \   DO 1080 M=1,5
C                                                                $SKIP END$
C  Zero the normal equations
C                                                              $SKIP START$
           DO 1040 I=1,8
                DO 1040 J=1,9
                EQN(I,J)=0.0D0
 1040      CONTINUE
C                                                                $SKIP END$
C  Form the normal equations
C                                                              $SKIP START$
           DO 1050 I=1,NFID
 1050      CALL ACCNEQ (CALCOR(1,I),CALCOR(2,I),OBSCOR(1,I),OBSCOR(2,I))
C                                                                $SKIP END$
C  Solve the normal equations
C                                                              $SKIP START$
           CALL LINSOL(8)
C                                                                $SKIP END$
C  Correct the approximate values of the transformation parameters
C                                                              $SKIP START$
           DO 1060 I=1,8
 1060      DEL(I)=DEL(I)+EQN(I,9)
C                                                                $SKIP END$
C  Test the solution for convergence
C                                                              $SKIP START$
           DO 1070 I=1,8
                D=DABS(DEL(I)/(DEL(I)-EQN(I,9))-1.0)
                IF (D.GT..001D0) GO TO 1080
 1070      CONTINUE
```

```
          RETURN
 1080 CONTINUE
C                                                              $SKIP END$
C                                                                   $END$
      END
C*********************************************************************
      SUBROUTINE LINSOL(NPAR)
C
C $CONFIG$="/T1 /LC"
C $NAME$
C          SUBROUTINE LINSOL
C $PATHS$
C          FUNCTIONS\ALL
C          MODULES\LINSOL
C $1$
C
C  Solution of (NPAR) linear equations in (NPAR) unknowns.
C                                                            $SKIP START$
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON CALCOR(2,50),OBSCOR(2,50),EQN(8,9),DEL(8),ICH3, NFID
C
      DO 1040 K=1,NPAR
           M=NPAR+1
           DO 1010 J=K,NPAR+1
                EQN(K,M)=EQN(K,M)/EQN(K,K)
 1010           M=M-1
           DO 1030 I=1,NPAR
                IF (I.EQ.K) GO TO 1030
                M=NPAR+1
                DO 1020 L=K,NPAR+1
                     EQN(I,M)=EQN(I,M)-EQN(I,K)*EQN(K,M)
 1020                M=M-1
 1030      CONTINUE
 1040 CONTINUE
C                                                              $SKIP END$
C                                                                   $END$
      END
C*********************************************************************
      SUBROUTINE ACCAPR (XG,YG,XP,YP)
C
C $CONFIG$="/T1 /LC"
C $NAME$
C          SUBROUTINE ACCAPR
C $PATHS$
C          FUNCTIONS\ALL
C          MODULES\ACCAPR
C $1$
C
C  Evaluate the contribution of one point to the 8 by 9 matrix of
C  normal equations for computation of approximate values of the
C  eight-parameter film shrinkage transformation.
C
C            XG:  Calibrated X Fiducial coordinate
C            YG:  Calibrated Y Fiducial coordinate
C            XP:    Observed X Fiducial coordinate
```

```
C          YP:    Observed Y Fiducial coordinate
C         EQN:  8 X 8 Coefficient matrix of the Normal Equation
C               with the vector of constants in column 9.
C                                                             $SKIP START$
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION AM(2,8), BM(2)
      COMMON CALCOR(2,50),OBSCOR(2,50),EQN(8,9),DEL(8),ICH3, NFID
C
      AM(1,1)=XP
      AM(1,2)=YP
      AM(1,3)=1.0D0
      AM(1,4)=-XG*XP
      AM(1,5)=-XG*YP
      AM(1,6)=0.0D0
      AM(1,7)=0.0D0
      AM(1,8)=0.0D0
      AM(2,1)=0.0D0
      AM(2,2)=0.0D0
      AM(2,3)=0.0D0
      AM(2,4)=-XP*YG
      AM(2,5)=-YP*YG
      AM(2,6)=XP
      AM(2,7)=YP
      AM(2,8)=1.0D0
      BM(1)=XG
      BM(2)=YG
      DO 1010 I=1,8
           DO 1010 J=1,8
           DO 1010 K=1,2
 1010      EQN(I,J)=EQN(I,J)+AM(K,I)*AM(K,J)
      DO 1020 I=1,8
           DO 1020 J=1,2
 1020      EQN(I,9)=EQN(I,9)+AM(J,I)*BM(J)
C                                                               $SKIP END$
C                                                                   $END$
      END
C**********
      SUBROUTINE ACCNEQ (XG,YG,XP,YP)
C
C $CONFIG$="/T1 /LC"
C $NAME$
C         SUBROUTINE ACCNEQ
C $PATHS$
C         FUNCTIONS\ALL
C         MODULES\ACCNEQ
C $1$
C
C  Evaluate the contribution of one point to the normal equation
C  required for Subroutine EIGHT.  The normal equations are
C  required to compute corrections to the last extimate of the
C  eight transformation parameters.  This is called once for each
C  point.
C
C          XG:  Calibrated X Fiducial coordinate
C          YG:  Calibrated Y Fiducial coordinate
```

```
C         XP:    Observed X Fiducial coordinate
C         YP:    Observed Y Fiducial coordinate
C        EQN:  8 X 8 Coefficient matrix of the Normal Equation
C              with the vector of constants in column 9.
C
C                                                            $SKIP START$
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION AM(2,2), BM(2,8), CM(2), AMM(2,2)
      COMMON CALCOR(2,50),OBSCOR(2,50),EQN(8,9),DEL(8),ICH3, NFID
C
      AM(1,1)=DEL(1)-XG*DEL(4)
      AM(1,2)=DEL(2)-XG*DEL(5)
      AM(2,1)=DEL(6)-YG*DEL(4)
      AM(2,2)=DEL(7)-YG*DEL(5)
      BM(1,1)=XP
      BM(1,2)=YP
      BM(1,3)=1.0D0
      BM(1,4)=-XP*XG
      BM(1,5)=-YP*XG
      BM(1,6)=0.0D0
      BM(1,7)=0.0D0
      BM(1,8)=0.0D0
      BM(2,1)=0.0D0
      BM(2,2)=0.0D0
      BM(2,3)=0.0D0
      BM(2,4)=-XP*YG
      BM(2,5)=-YP*YG
      BM(2,6)=XP
      BM(2,7)=YP
      BM(2,8)=1.0D0
      CM(1)=XP*AM(1,1)+YP*AM(1,2)+DEL(3)-XG
      CM(2)=XP*AM(2,1)+YP*AM(2,2)+DEL(8)-YG
C                                                              $SKIP END$
C  Form modified covariance matrix AMM
C                                                            $SKIP START$
      DO 1010 I=1,2
           DO 1010 J=1,2
           AMM(I,J)=0.0D0
           DO 1010 K=1,2
           AMM(I,J)=AMM(I,J)+AM(I,K)*AM(J,K)
 1010 CONTINUE
      D=AMM(1,1)*AMM(2,2)-AMM(1,2)*AMM(2,1)
      AM(1,1)=AMM(2,2)/D
      AM(2,2)=AMM(1,1)/D
      AM(1,2)=-AMM(2,1)/D
      AM(2,1)=AMM(1,2)
C                                                              $SKIP END$
C  Form normal equations
C                                                            $SKIP START$
      DO 1020 I=1,8
           DO 1020 J=1,8
           DO 1020 K=1,2
           DO 1020 L=1,2
           EQN(I,J)=EQN(I,J)+BM(K,I)*AM(K,L)*BM(L,J)
 1020 CONTINUE
```

```
      DO 1030 I=1,8
            DO 1030 K=1,2
            DO 1030 L=1,2
            EQN(I,9)=EQN(I,9)-BM(K,I)*AM(K,L)*CM(L)
 1030 CONTINUE
      RETURN
C                                                                  $SKIP END$
C                                                                       $END$
      END
C**********
      SUBROUTINE INVERT (A,N,D,L,M)
C
C $CONFIG$="/T1 /LC"
C $NAME$
C           SUBROUTINE INVERT
C $PATHS$
C           FUNCTIONS\ALL
C           MODULES\INVERT
C $1$
C
C  Find the Inverse of a Matrix by the Gaussian Elimination Method.
C  A: Array in which the matrix to be inverted is located.
C     The routine will search for the largest non-singular matrix in
C     the array A and invert it & return it in the same locations of A.
C  N: The first dimension of A. It must be a variable in the call list.
C     The rank of largest matrix contained in A will be returned in N.
C  D: The determinant of the largest non-singular matrix in A.
C  L & M: Vectors of dimension N used temporarily.
C
C                                                                $SKIP START$
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION A(400), L(20), M(20)
C                                                                  $SKIP END$
C  Initiate the continued product of pivots becoming the determinant.
C                                                                $SKIP START$
      D=1.0D0
C                                                                  $SKIP END$
C  Initiate the counter which contains the rank of the matrix.
C                                                                $SKIP START$
      KSAVE=0
C                                                                  $SKIP END$
C  Start the main elimination loop.
C                                                                $SKIP START$
      DO 1090 K=1,N
C                                                                  $SKIP END$
C  Search for the largest element
C                                                                $SKIP START$
            L(K)=K
            M(K)=K
            KK=K+N*(K-1)
            BIGA=A(KK)
            DO 1010 I=K,N
                  DO 1010 J=K,N
                  IJ=I+N*(J-1)
                  IF (DABS(BIGA).GE.DABS(A(IJ))) GO TO 1010
```

```
          BIGA=A(IJ)
          L(K)=I
          M(K)=J
 1010   CONTINUE
C                                                              $SKIP END$
C  Largest element of zero means the largest matrix in A is less than N.
C                                                            $SKIP START$
        IF (BIGA.EQ.0) GO TO 1100
C                                                              $SKIP END$
C  Interchange rows
C                                                            $SKIP START$
        J=L(K)
        KSAVE=K
        IF (L(K).LE.K) GO TO 1030
        DO 1020 I=1,N
             KI=K+N*(I-1)
             JI=J+N*(I-1)
 1020   CONTINUE
C                                                              $SKIP END$
C  Interchange columns.
C                                                            $SKIP START$
 1030   I=M(K)
        IF (M(K).LE.K) GO TO 1050
        DO 1040 J=1,N
             JK=J+N*(K-1)
             JI=J+N*(I-1)
 1040   CONTINUE
C                                                              $SKIP END$
C  Divide column by minus pivot
C                                                            $SKIP START$
 1050   DO 1060 I=1,N
             IF (I.EQ.K) GO TO 1060
             IK=I+N*(K-1)
             A(IK)=A(IK)/(-A(KK))
 1060   CONTINUE
C                                                              $SKIP END$
C  Reduce matrix
C                                                            $SKIP START$
        DO 1070 I=1,N
             DO 1070 J=1,N
             IF (I.EQ.K.OR.J.EQ.K) GO TO 1070
             IJ=I+N*(J-1)
             IK=I+N*(K-1)
             KJ=K+N*(J-1)
             A(IJ)=A(IK)*A(KJ)+A(IJ)
 1070   CONTINUE
C                                                              $SKIP END$
C  Divide row by pivot
C                                                            $SKIP START$
        DO 1080 J=1,N
             IF (J.EQ.K) GO TO 1080
             KJ=K+N*(J-1)
             A(KJ)=A(KJ)/A(KK)
 1080   CONTINUE
C                                                              $SKIP END$
```

```
C  Continued product of pivots
C                                                       $SKIP START$
              D=D*A(KK)
              A(KK)=1.0D0/A(KK)
 1090 CONTINUE
C                                                         $SKIP END$
C  Final row and column interchange
C                                                       $SKIP START$
 1100 K=KSAVE+1
 1110 K=K-1
      IF (K.LE.0) GO TO 1150
C                                                         $SKIP END$
C  Restore columns.
C                                                       $SKIP START$
      I=L(K)
      IF (I.LE.K) GO TO 1130
      DO 1120 J=1,N
          JK=J+N*(K-1)
          JI=J+N*(I-1)
 1120     A(JK)=-A(JI)
C                                                         $SKIP END$
C  Restore rows.
C                                                       $SKIP START$
 1130 J=M(K)
      IF (J.LE.K) GO TO 1110
      DO 1140 I=1,N
          KI=K+N*(I-1)
          JI=J+N*(I-1)
 1140     A(KI)=-A(JI)
      GO TO 1110
C                                                         $SKIP END$
C  Set the rank of the matrix and return to the calling routine.
C                                                       $SKIP START$
 1150 RETURN
C                                                         $SKIP END$
C                                                             $END$
      END
```

# PC Giant

## Source Code

## File Name: TPLATE.FOR

## (T-Plate Constraint Program For GIANT)

## 14 June 1990

```
      DIMENSION A(3), B(3), O(3)
      character*8 cx
      open(10, file='obj.out', status='old')
      read(10, *)cx, o, cx, a, cx, b
C   CEN,  LFT, RT
      CALL TPLATE(O, A, B)
      WRITE (*, *) O, A, B
      END


      SUBROUTINE TPLATE(O, A, B)
      DIMENSION A(3), B(3), O(3), C(3), U(3), V(3)
C     C is original center of A & B & then adjusted to 62.906mm.
C     U=A X B then adjusted for perpendicular distance of 62.860mm.
C     V=U X C is vector from C to A & -V is from C to B.
      CT=0.
      DO 20 I=1, 3
      A(I)=A(I)-O(I)
      B(I)=B(I)-O(I)
      C(I)=(A(I)+B(I))/2
   20 CT=CT+C(I)**2
      CT=SQRT(CT)
      DO 30 I=1, 3
   30 C(I)=C(I)/CT*.062906
      U(1)=A(2)*B(3)-A(3)*B(2)
      U(2)=A(3)*B(1)-A(1)*B(3)
      U(3)=A(1)*B(2)-A(2)*B(1)
      UT=SQRT(U(1)**2+U(2)**2+U(3)**2)
      DO 40 I=1, 3
   40 U(I)=U(I)/UT*.99926875
      V(1)=U(2)*C(3)-U(3)*C(2)
      V(2)=U(3)*C(1)-U(1)*C(3)
      V(3)=U(1)*C(2)-U(2)*C(1)
      DO 50 I=1, 3
      A(I)=C(I)-V(I)+O(I)
   50 B(I)=C(I)+V(I)+O(I)
      RETURN
      END
```

# PC Giant/Prep

## Subroutine Flow Diagrams

14 June 1990

Program: GIANT

DIAGRAM'er v2.1

Run: 06/15/1990 16:42:35
Page 1 of Diagram No. 1

```
Program    GIANT
└─P.GIANT
   ├─S.CLR
   ├─S.TOPLFT
   ├─S.CURDWN
   ├─S.BEEP
   ├─S.PHASE1
   │  ├─S.RDFRAM
   │  │  ├ C/TAPES/
   │  │  ├─S.NEWPAG
   │  │  ├─S.LISTTP
   │  │  │  ├─S.CLR
   │  │  │  └─S.TOPLFT
   │  │  ├─S.READIM
   │  │  │  ├ C/TAPES/
   │  │  │  ├─S.TSTFRM
   │  │  │  ├─S.REFRM
   │  │  │  ├─S.CLR
   │  │  │  ├─S.TOPLFT
   │  │  │  ├─S.CURDWN
   │  │  │  └─S.BEEP
   │  │  ├─S.TSTFRM
   │  │  ├─S.REFRM
   │  │  ├─S.GETFR
   │  │  │  ├─S.CLR
   │  │  │  ├─S.TOPLFT
   │  │  │  ├─S.CURDWN
   │  │  │  ├─S.BEEP
   │  │  │  └─s.GETPT
   │  │  ├─S.CLR
   │  │  ├─S.TOPLFT
   │  │  ├─S.CURDWN
   │  │  ├─S.BEEP
   │  │  ├─F.DEGRAD
   │  │  │  ├─S.CLR
   │  │  │  ├─S.TOPLFT
   │  │  │  ├─S.CURDWN
   │  │  │  └─S.BEEP
   │  │  ├─S.RADDEG
   │  │  └─s.GETPT
   │  ├─S.BLOCKD
   │  │  └─S.SORT
   │  └─S.MERGEG
   │     ├─S.SORT
   │     └─S.PRINTM
   │
```

```
  │        ├ C/TAPES/
  │        └─S.NEWPAG
  ├─S.PHASE2
  │  ├─S.FILL
  │  ├─S.INITID
  │  │  ├─s.DROPID
  │  │  └─S.LOCTID
  │  │     ├─S.CLR
  │  │     ├─S.TOPLFT
  │  │     ├─S.CURDWN
  │  │     └─S.BEEP
  │  ├─S.MISCOM
  │  │  ├ C/WORK23/
  │  │  ├─S.INITID
  │  │  │  ├─s.DROPID
  │  │  │  └─S.LOCTID
  │  │  │     ├─S.CLR
  │  │  │     ├─S.TOPLFT
  │  │  │     ├─S.CURDWN
  │  │  │     └─S.BEEP
  │  │  ├─S.MODID
  │  │  │  ├─S.CLR
  │  │  │  ├─S.TOPLFT
  │  │  │  ├─S.CURDWN
  │  │  │  └─S.BEEP
  │  │  ├─S.LOCTID
  │  │  │  ├─S.CLR
  │  │  │  ├─S.TOPLFT
  │  │  │  ├─S.CURDWN
  │  │  │  └─S.BEEP
  │  │  ├─S.ROTMAT
  │  │  │  ├─S.COPY
  │  │  │  └─S.MPYAB
  │  │  ├─S.PLHXYZ
  │  │  ├─S.COPY
  │  │  ├─s.DROPID
  │  │  ├─S.FILL
  │  │  ├─S.MPYABT
  │  │  ├─S.MPYAB
  │  │  ├─S.MPYATB
  │  │  ├─S.ADDMAT
  │  │  ├─S.INVRT
  │  │  └─S.XYZPLH
  │  ├─S.NEWPAG
```

```
 │  ├─S.LEASTQ
 │  │  ├ C/WORK23/
 │  │  ├─S.FILL
 │  │  ├─S.STSUBM
 │  │  │  ├─S.MODID
 │  │  │  │  ├─S.CLR
 │  │  │  │  ├─S.TOPLFT
 │  │  │  │  ├─S.CURDWN
 │  │  │  │  └─S.BEEP
 │  │  │  ├─S.LOCTID
 │  │  │  │  ├─S.CLR
 │  │  │  │  ├─S.TOPLFT
 │  │  │  │  ├─S.CURDWN
 │  │  │  │  └─S.BEEP
 │  │  │  ├─S.TRANSP
 │  │  │  ├─S.ADDMAT
 │  │  │  └─S.COPY
 │  │  ├─S.STSUBV
 │  │  │  ├─S.MODID
 │  │  │  │  ├─S.CLR
 │  │  │  │  ├─S.TOPLFT
 │  │  │  │  ├─S.CURDWN
 │  │  │  │  └─S.BEEP
 │  │  │  ├─S.LOCTID
 │  │  │  │  ├─S.CLR
 │  │  │  │  ├─S.TOPLFT
 │  │  │  │  ├─S.CURDWN
 │  │  │  │  └─S.BEEP
 │  │  │  ├─S.ADDMAT
 │  │  │  └─S.COPY
 │  │  ├─S.LOCTID
 │  │  │  ├─S.CLR
 │  │  │  ├─S.TOPLFT
 │  │  │  ├─S.CURDWN
 │  │  │  └─S.BEEP
 │  │  ├─S.ROTMAT
 │  │  │  ├─S.COPY
 │  │  │  └─S.MPYAB
 │  │  ├─S.PLHXYZ
 │  │  ├─S.COPY
 │  │  ├─S.DROP
 │  │  │  ├ C/WORK23/
 │  │  │  ├─S.STSUBM
 │  │  │  │  ├─S.MODID
```

```
|    |    |    |    |    ├─S.CLR
|    |    |    |    |    ├─S.TOPLFT
|    |    |    |    |    ├─S.CURDWN
|    |    |    |    |    └─S.BEEP
|    |    |    |    ├─S.LOCTID
|    |    |    |    |    ├─S.CLR
|    |    |    |    |    ├─S.TOPLFT
|    |    |    |    |    ├─S.CURDWN
|    |    |    |    |    └─S.BEEP
|    |    |    |    ├─S.TRANSP
|    |    |    |    ├─S.ADDMAT
|    |    |    |    └─S.COPY
|    |    |    ├─S.INVRT
|    |    |    ├─S.STSUBV
|    |    |    |    ├─S.MODID
|    |    |    |    |    ├─S.CLR
|    |    |    |    |    ├─S.TOPLFT
|    |    |    |    |    ├─S.CURDWN
|    |    |    |    |    └─S.BEEP
|    |    |    |    ├─S.LOCTID
|    |    |    |    |    ├─S.CLR
|    |    |    |    |    ├─S.TOPLFT
|    |    |    |    |    ├─S.CURDWN
|    |    |    |    |    └─S.BEEP
|    |    |    |    ├─S.ADDMAT
|    |    |    |    └─S.COPY
|    |    |    ├─S.MPYAB
|    |    |    └─S.MPYABT
|    |    ├─S.CONEQN
|    |    |    ├ C/COEFF/
|    |    |    ├─S.LOCTID
|    |    |    |    ├─S.CLR
|    |    |    |    ├─S.TOPLFT
|    |    |    |    ├─S.CURDWN
|    |    |    |    └─S.BEEP
|    |    |    ├─S.REFRCT
|    |    |    |    ├─S.MPYAB
|    |    |    |    └─S.MPYATB
|    |    |    ├─S.SUBMAT
|    |    |    ├─S.MPYATB
|    |    |    ├─S.MPYAB
|    |    |    ├─S.COPY
|    |    |    └─S.ADDMAT
|    |    ├─S.MPYATB
```

```
│    │    ├─S.ADDMAT
│    │    ├─S.INVRT
│    │    └─S.MPYAB
│    ├─S.BACKSL
│    │    ├ C/WORK23/
│    │    ├─S.MPYATB
│    │    ├─S.SUBMAT
│    │    └─S.MPYAB
│    ├─S.UPDATG
│    │    ├ C/WORK23/
│    │    ├─S.CLR
│    │    ├─S.TOPLFT
│    │    ├─S.CURDWN
│    │    ├─S.BEEP
│    │    ├─S.MPYAB
│    │    └─S.SUBMAT
│    ├─S.CLR
│    ├─S.TOPLFT
│    ├─S.CURDWN
│    ├─S.BEEP
│    ├─S.RADDEG
│    ├─F.PAKDMS
│    ├─S.LSTPLR
│    │    ├ C/TAPES/
│    │    ├─S.INITID
│    │    │    ├─s.DROPID
│    │    │    └─S.LOCTID
│    │    │         ├─S.CLR
│    │    │         ├─S.TOPLFT
│    │    │         ├─S.CURDWN
│    │    │         └─S.BEEP
│    │    ├─S.MODID
│    │    │    ├─S.CLR
│    │    │    ├─S.TOPLFT
│    │    │    ├─S.CURDWN
│    │    │    └─S.BEEP
│    │    ├─S.LOCTID
│    │    │    ├─S.CLR
│    │    │    ├─S.TOPLFT
│    │    │    ├─S.CURDWN
│    │    │    └─S.BEEP
│    │    ├─S.ROTMAT
│    │    │    ├─S.COPY
│    │    │    └─S.MPYAB
```

```
│    │    ├─S.PLHXYZ
│    │    ├─S.COPY
│    │    ├─s.DROPID
│    │    ├─S.SUBMAT
│    │    ├─S.MPYATB
│    │    ├─S.REFRCT
│    │    │    ├─S.MPYAB
│    │    │    └─S.MPYATB
│    │    └─S.NEWPAG
│    └─S.PERROR
│         ├ C/WORK22/
│         ├─S.STSUBM
│         │    ├─S.MODID
│         │    │    ├─S.CLR
│         │    │    ├─S.TOPLFT
│         │    │    ├─S.CURDWN
│         │    │    └─S.BEEP
│         │    ├─S.LOCTID
│         │    │    ├─S.CLR
│         │    │    ├─S.TOPLFT
│         │    │    ├─S.CURDWN
│         │    │    └─S.BEEP
│         │    ├─S.TRANSP
│         │    ├─S.ADDMAT
│         │    └─S.COPY
│         ├─S.MPYABT
│         ├─S.COPY
│         ├─S.MPYAB
│         └─S.SUBMAT
└─S.PHASE3
     ├ C/WORK31/
     ├─S.SRTGPS
     │    ├ C/TAPES/
     │    └ C/WORK21/
     ├─S.LSTPNH
     │    ├ C/TAPES/
     │    ├ C/WORK31/
     │    ├ C/ANTHR/
     │    ├─S.NEWPAG
     │    ├─S.RADDEG
     │    ├─S.TRED2
     │    ├─S.TQL2
     │    │    └─F.PYTHAG
     │    ├─F.PAKDMS
```

```
    │    └─S.STUFFP
    │         ├ C/TAPES/
    │         └ C/ANTHR/
    ├─S.LSTGRS
    │    ├ C/TAPES/
    │    ├─S.SORTER
    │    │    ├─S.CLR
    │    │    ├─S.BEEP
    │    │    └─S.CURDWN
    │    ├─S.NEWPAG
    │    └─S.RADDEG
    └─S.ANTHRO
         ├ C/TAPES/
         ├ C/ANTHR/
         ├─S.NEWPAG
         └─S.NBDL
              ├ C/ANTHR/
              └─S.UVEC
```

ogram: MAIN

AGRAM'er v2.1 Run: Ø6/Ø7/199Ø 12:2Ø:54



ge 1 of Diagram No. 1

```
          Program     MAIN

—P.MAIN
   ├ C/-BLANK/
   ├—S.FOURP
   │   ├ C/-BLANK/
   │   └—S.LINSOL
   │       └ C/-BLANK/
   ├—S.FIVEP
   │   ├ C/-BLANK/
   │   └—S.LINSOL
   │       └ C/-BLANK/
   ├—S.SIXP
   │   ├ C/-BLANK/
   │   └—S.INVERT
   └—S.EIGHTP
       ├ C/-BLANK/
       ├—S.ACCAPR
       │   └ C/-BLANK/
       ├—S.LINSOL
       │   └ C/-BLANK/
       └—S.ACCNEQ
           └ C/-BLANK/
```

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



**1. AGENCY USE ONLY** *(Leave Blank)*

**2. REPORT DATE**
June 1990

**3. REPORT TYPE AND DATES COVERED**
Interim

**4. TITLE AND SUBTITLE**

Evaluation of the Anthropometry System

**5. FUNDING NUMBERS**

N00205-90-M-D005

**6. AUTHOR(S)**

Michael E. Pittman, Ph.D.

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Naval Biodynamics Laboratory
P. O. Box 29407
New Orleans, LA 70189-0407

**8. PERFORMING ORGANIZATION REPORT NUMBER**

NBDL-93R007

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Naval Medical Research and Development Command
National Naval Medical Center
Building 1, Tower 12
Bethesda, MD 20889-5044

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

This publication provides documentation on the evaluation of the present X-ray Anthropometry System of the Naval Biodynamics Laboratory. Analysis, exposition, enhancement and documentation of the x-ray digitization and 3-D reconstruction algorithms, development of error analysis code for incorporation into digitization and reconstruction algorithms.

**14. SUBJECT TERMS**
X-Ray Anthropometry, PREP and GIANT, Camera Calibration, and Vector Constraints.

**15. NUMBER OF PAGES**
105

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | |